NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

EXTENDING THE STATE-OF-THE-ART FOR THE COMAN / ATCAL METHODOLOGY

by

Ugur Ziya Yildirim

March, 1999

Thesis Advisor:	James G. Taylor
Thesis Co-Advisor:	William S. Murphy Jr.
Second Reader:	Ronald L. Brown

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1999	Master's Thesis

4. TITLE AND SUBTITLE EXTENDING THE STATE-OF-THE ART FOR THE COMAN ATCAL METHODOLOGY	5. FUNDING NUMBERS
6. AUTHOR(S) Yildirim, Ugur Z.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
Approved for public release; distribution is unlimited.	

This thesis reviews currently existing attrition methodologies and critically examines the (ATtrition CALibration) ATCAL approach for heterogeneous force mixes. Lanchester attrition-rate coefficients must be calibrated for use in ATCAL. When the aggregated combat results from a Lanchester-type attrition model agree with the results of a high-resolution simulation, for a particular mix of weapons on either side, the equation is said to be calibrated. Combat scenario runs are made in the high-resolution JANUS combat model. A maximum likelihood estimation approach (COMAN), which incorporates target priority and availability information, is used to estimate the Lanchester attrition rates in the ATCAL model. A continuous-time, three-state Markov chain model of target acquisition in JANUS is used to obtain the target availability parameters required in the COMAN approach. Bootstrap confidence intervals are developed for the attrition-rate estimates and the target availability parameters. The ability of the ATCAL methodology to replay JANUS results in an aggregated, heterogeneous-force replay model is investigated. These developments have the potential for allowing high-resolution simulation results to be extrapolated to a wider spectrum of conditions (e.g. force mixes) than currently possible by virtue of the more detailed nature of the aggregated-replay model involved.

14. SUBJECT TERMS JANUS, ATCAL, COMAN, MLE, Combat Models, Simulation, Attrition	15. NUMBER OF PAGES 143
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFI- CATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited

EXTENDING THE STATE-OF-THE-ART FOR THE COMAN / ATCAL METHODOLOGY

Ugur Ziya Yildirim
First Lieutenant, Turkish Army
B.S., United States Military Academy, 1991

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESERACH

from the

NAVAL POSTGRADUATE SCHOOL
March 1999

ABSTRACT

This thesis reviews currently existing attrition methodologies and critically examines the (ATtrition CALibration) ATCAL approach for heterogeneous force mixes. Lanchester attrition-rate coefficients must be calibrated for use in ATCAL. When the aggregated combat results from a Lanchester-type attrition model agree with the results of a high-resolution simulation, for a particular mix of weapons on either side, the equation is said to be calibrated. Combat scenario runs are made in the high-resolution JANUS combat model. A maximum likelihood estimation approach (COMAN), which incorporates target priority and availability information, is used to estimate the Lanchester attrition rates in the ATCAL model. A continuous-time, three-state Markov chain model of target acquisition in JANUS is used to obtain the target availability parameters required in the COMAN approach. Bootstrap confidence intervals are developed for the attrition-rate estimates and the target availability parameters. The ability of the ATCAL methodology to replay JANUS results in an aggregated, heterogeneous-force replay model is investigated. These developments have the potential for allowing high-resolution simulation results to be extrapolated to a wider spectrum of conditions (e.g. force mixes) than currently possible by virtue of the more detailed nature of the aggregated-replay model involved.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

Essentially all current theater-level air-ground models and joint-warfare models play aggregated ground combat and assess ground losses by means of the ATtrition CALibration (ATCAL) method developed by the United States (U.S.) Army Center for Army Analysis (CAA) in the early 1980s. In fact, essentially the same attrition algorithm (developed originally by CAA) is used in all these models. However, there is no documentation of the theoretical basis for ATCAL. Thus, there is a need to document the theoretical basis of this important approach for determining Lanchester-type attrition-rate coefficients for large- scale ground-combat models. This knowledge can be used to improve ATCAL to provide it with the ability to make better extrapolations and be more responsive to the new analysis requirements of the Twenty-First Century. Thus, the methodological developments of this thesis can have a direct impact on DoD tools currently used for investigating issues involving large-scale ground forces.

Almost all DoD models involving large-scale ground forces assess attrition for these ground forces by means of algorithms based on Lanchester-type equations of warfare. There are two main approaches for determining numerical values for the attrition-rate coefficients in such differential-equation based combat models

(1) Bonder-Farrell Approach,

(2) ATCAL Approach.

The ATCAL approach, whose extension is the subject of the thesis at hand, estimates parameter values of an aggregated-replay model from the output of a high-resolution, Monte-Carlo combat simulation. This estimation phase is called ATCAL

Phase I. The replay of high-resolution simulation results by an aggregated-replay model is called ATCAL Phase II. ATCAL evolved from a seminal work called COMAN. A maximum likelihood estimation technique was used in COMAN to develop statistical estimates for these parameter values under an extremely restrictive condition (namely, all enemy target types have exactly the same availability for engagement by each opposing firer type). CAA could not retain such a restrictive condition for the U.S. Army studies. Unfortunately, this has also meant that the estimation of ATCAL parameter values based on any type of theoretically-sound statistical theory also was also not retained.

This thesis extends ATCAL methodology by developing a submodel to determine the availability of each enemy target type to any opposing firer type by consideration of a continuous-time, three-state Markov chain. Target availability is the probability that a particular enemy target type is visible and acquired by a given opposing firer type. The calculation of transition rates for this Markov chain model requires more detailed output data involving line-of-sight and target-acquisition information from a high-resolution Monte-Carlo combat simulation than is used in current practice. Target availability is computed as a function of these transition rates by the use of a JAVA program written to reduce simulation output data. With values obtained for all target availabilities, it is a straightforward matter to compute maximum likelihood estimates for Lanchester attrition-rate coefficients, the rate at which a given firer type kills acquired enemy targets of a given type.

The theory developed in this thesis is implemented in a case study involving the JANUS model, a contemporary, high-resolution Monte-Carlo simulation extensively

used by the U.S. Army for both training and analysis. A major difficulty that was encountered in practice was to obtain the more-detailed line-of-sight data from JANUS. This difficulty was overcome by modifying the source code for the JANUS program to force JANUS to output the required information in a text file.

After the parameters of the ATCAL equations have been obtained as a result of the data reduction from the JANUS simulation model, the ability of the ATCAL Phase II methodology to replay the high-resolution-combat-simulation results in an aggregated, heterogeneous-force replay model is investigated.

These theoretical developments have the potential for allowing such high-resolution results to be extrapolated to a wider spectrum of conditions (e.g. varying force mixes) than is currently possible by virtue of the more detailed nature of the aggregated-replay model involved. Future work should computationally investigate the playing of artillery effects. Extension and implementation of the ideas developed in this thesis should materially improve the state-of-the-art for aggregated, force-on-force models.

ACKNOWLEDGEMENTS

I am grateful to Professor James G. Taylor, MAJ William S. Murphy and CDR. Ronald L. Brown for their contributions, continuous guidance, and support in carrying out this research.

A special thanks goes to MAJ LeRoy "Jack" Jackson for the many long hours he spent with me in developing the JAVA program used in data reduction of detection files obtained from JANUS.

I would like to thank Mr. Harold Yamuchi of TRAC/Monterey for his help in modifying the source code for the JANUS program to obtain information necessary for this research.

I would like to thank Professor Gordon Clark for his generosity in sharing his ideas and insight into combat modeling.

Most importantly, I am indebted to the Turkish Army and the Turkish Republic for providing me with the opportunity to pursue a Master's degree.

I. INTRODUCTION

A. OVERVIEW

The United States (U.S.) Department of Defense (DoD) and other ministries of defense often rely on computer-based models and simulations to support decision making. These models and simulations are used in a wide variety of application areas, which include material acquisition, logistical, and combat operations. This thesis focuses on aggregated combat models and simulations that are used to provide insight into the potential outcomes of proposed courses of actions prior to committing members of the military into harm's way.

Lanchester-type equations [Ref. 1] play a central role in representing ground-war attrition in aggregated combat models. These aggregated models use a variety of methods to determine scenario or equipment dependent Lanchester attrition-rate coefficients that represent the rate at which an individual firer kills enemy targets. A major problem in applying any type of Lanchester-type model to a real-world military problem is the estimation of the numerical value for a Lanchester attrition-rate coefficient. In the homogeneous case, where each of the opposing forces only has one type of equipment, the development of the Lanchester attrition-rate coefficients is fairly trivial. However, within the context of heterogeneous forces that participate in combined arms combat, the estimation of Lanchester attrition-rate coefficients can be a formidable task. This thesis

examines the methods that are used to develop the Lanchester attrition-rate coefficients for both homogeneous and heterogeneous forces.

B. BACKGROUND

The military modeling community currently uses the Bonder-Farrell approach [Ref. 2] and the ATtrition CALibration (ATCAL) approach [Ref. 3] to determine the Lanchester attrition-rate coefficients. These two methods approach the problem from completely different perspectives. The Bonder-Farrell approach uses an analytical sub-model that is independent of any high-resolution simulation model to estimate the Lanchester attrition-rate coefficients. [Ref. 2] The ATCAL approach estimates the Lanchester attrition-rate coefficients from the output of a high-resolution simulation model. [Ref. 3] This thesis focuses on the ATCAL approach. The Concepts Evaluation Model (CEM) [Ref. 4], the Tactical Warfare (TACWAR 4.1) Model [Ref. 5] and the Ground Combat module of the Integrated Theater Engagement Model (ITEM) [Ref. 6] use the ATCAL approach to determine the attrition of aggregated forces.

ATCAL has two phases. In ATCAL Phase I, the Lanchester attrition-rate coefficients are estimated from the output of a high-resolution Monte-Carlo combat simulation. In ATCAL Phase II, a fitted parameter analytical model uses a set of Lanchester-type [Ref. 1] equations and the Lanchester attrition-rate coefficients to simulate force attrition. [Ref. 3]

The primary user of the ATCAL method is the Center for Army Analysis (CAA). Their implementation of ATCAL Phase I uses the output of the Combat Sample

Generator (COSAGE) model to obtain the Lanchester attrition-rate coefficients [Ref. 5]. COSAGE is a two-sided, stochastic, high-resolution combat model that simulates forty-eight hours of conventional combat between two opposing forces over a variety of defensive and offensive tactical combat situations [Ref. 5]. Their implementation of ATCAL Phase II uses CEM as its higher level aggregate model. [Ref. 4]

Lanchester attrition-rate coefficients must be calibrated for use in ATCAL. When the aggregated combat results from a Lanchester-type [Ref. 1] attrition model agree with the results of a high-resolution simulation, for a particular mix of weapons on either side, the equation is said to be calibrated [Ref. 3].

Taylor notes several concerns about the current ATCAL methodology used by the CAA. There is essentially no documentation of the theoretical basis of ATCAL. There is an ad hoc estimation of model parameters from the high-resolution output in that only the final attrition results are considered. There is no time stamping of data. [Ref. 2]

This thesis demonstrates the application of an alternative method for the estimation of Lanchester attrition-rate coefficients used in ATCAL. The CAA could potentially use this alternative method demonstrated in this thesis if COSAGE source code could be modified to provide some additional data points as explained in Chapter IV.

C. PURPOSE AND RATIONALE

The purpose of this study is to propose and demonstrate the application of a sound mathematical formulation and a maximum likelihood estimation (MLE) approach to

determine the Lanchester attrition rates for direct-fire engagements of heterogeneous force mixes. This is called ATCAL Phase I by the CAA. Replay of the high-resolution simulation results by aggregate force equations using the parameters obtained from ATCAL Phase I is called ATCAL Phase II.

In ATCAL Phase I, this study uses the COMAN MLE approach. COMAN stands for COMbat ANalysis, a model that was first developed by Gordon M. Clark in his Ph.D. thesis in 1969 [Ref. 7]. General Research Corporation made some improvements on the COMAN model and called it the COMAN Extended (COMANEX) model. COMANEX became the predecessor to ATCAL. [Ref. 8].

In his COMAN model, Clark estimated Lanchester attrition-rate coefficients from a high-resolution simulation of battalion-sized combat engagements of heterogeneous forces using MLEs. Clark also introduced target priorities and target availability parameters when he extended his homogeneous-force model to describe attrition for heterogeneous forces. Each weapon type was ranked with a priority for the firers of the opposing side. Clark assumed that this priority ranking held for all firers on the opposing force and that each firer allocated his fire on the highest priority target acquired. Clark developed MLEs for target availability parameters and the Lanchester attrition-rate coefficients. The reason for target availability parameters is that one needs to consider the probability of whether a particular target type can actually be engaged by a given firer type. [Ref. 7]

This thesis uses Taylor's adaptation of Clark's equations to estimate Lanchester attrition-rate coefficients. [Ref. 2] JANUS, a high-resolution simulation, [Ref. 9] is used

to generate time-stamped attrition and detection data to obtain model parameter estimates. JANUS is preferred over the COSAGE model for use in this thesis for several reasons. First, it is readily available at the Naval Postgraduate School. Second, the author plans to take the results of this study and use them in analysis at the JANUS site in Turkey. Third, JANUS provides detailed target line-of-sight (LOS) and detection information. [Ref. 9]

This study exploits JANUS' LOS and detection information to compute target availability information for all different weapon types used in Taylor's adaptation of Clark's heterogeneous force equations. Clark used complicated maximum likelihood estimator equations to estimate target availability and assumed target availability was the same for all different heterogeneous target types in his model.

Rather than use Clark's existing methodology, this study uses a modified version of a continuous-time, three-state Markov chain model of target acquisition to obtain different target availability parameters for each weapon system. This approach was considered by Clark but never used in his thesis. Thus, the use of a Markov chain model of target acquisition in JANUS to obtain different target availability values for all different weapon types is an extension to Clark's model [Ref. 7].

The source code for the JANUS program had to be modified to output the times when LOS between two entities in the simulation were lost. These results along with other detection information are output to a text file. A JAVA program had to be written for this thesis to extract the necessary information from JANUS' modified detection files for use in the Markov chain model of target acquisition. Additionally, another JAVA program is written to extract the necessary information for heterogeneous force equations,

i.e., the times between kills of entities in the simulation and the number of entities remaining at any given time in the simulation, from JANUS' modified direct-fire kill files.

This study also attempts to document the theoretical basis of the ATCAL methodology. In Taylor's many visits to the CAA, he has been unable to locate any report, paper, or notes documenting the theoretical basis of ATCAL. His discussions with several technical people at CAA revealed that no such documentation has ever existed. [Ref. 10]

To summarize, this study involves the following specific tasks:

1. Review the existing attrition methodologies for aggregated forces in general, and provide a through documentation of the theoretical basis of the ATCAL methodology.
2. Perform an ATCAL Phase I analysis by applying a sound MLE approach to time-stamped attrition and acquisition data obtained from JANUS simulation runs using a heterogeneous force scenario. The attrition rate parameters obtained here will be used in an aggregate model. Write two JAVA programs to extract information necessary for the COMAN MLE from JANUS' modified detection and direct-fire kill files.
3. Use a three-state, continuous-time Markov chain model of target acquisition to estimate the target availability parameters contained in the heterogeneous-force ATCAL equations. Use the JAVA program written to be used with

JANUS' detection files to compute the rates and the steady state solution of the Markov chain model.

4. Perform an ATCAL Phase II analysis by running an aggregate attrition model in an EXCEL spreadsheet that uses ATCAL equations to determine the attrition of forces.

D. SCOPE AND LIMITATIONS

A detailed discussion of other existing attrition methodologies is beyond the scope of this study. Instead, introductory information is provided simply to acquaint the reader with other currently existing methodologies. This study only deals with the ATCAL approach for the direct-fire engagements of heterogeneous forces. An analysis of the ATCAL approach for attrition resulting from indirect fire weapons, such as mortars and artillery, is beyond the scope of this study.

Existing attrition methodologies that are discussed include the Force Ratio approach [Ref. 1] and Lanchester-type [Ref. 1] models. The Lanchester-type [Ref. 1] models include both the Bonder-Farrell approach and the ATCAL approach. These attrition methodologies are discussed in Chapter II with an emphasis on the ATCAL methodology. In Chapter III, the target acquisition process in the JANUS combat model is addressed in detail as it relates to this research. The COMAN MLE approach and the method of obtaining target availability values in the COMAN MLE by using a three-state, continuous-time Markov chain model are discussed in Chapter IV. The analysis of Phase

I and Phase II results are presented in Chapter V. Finally, conclusions and recommendations are presented in Chapter VI.

II. ATTRITION METHODOLOGIES

This chapter provides introductory information on some of the currently used attrition methodologies. The Force Ratio approach is discussed first, followed by two Lanchester-type models, which use the Bonder-Farrell and the ATCAL approaches. A detailed discussion on the theoretical basis of the ATCAL methodology is also presented. ATCAL methodology is the main focus of this study.

A. FORCE RATIO APPROACH

In this approach, the heterogeneous forces on one side are aggregated into a single equivalent homogeneous force by using an index-number method [Ref. 1]. More specifically, each weapon system is assigned a score based on a common weighting criteria. Summing over the number of systems multiplied by their respective scores generates an overall firepower index (FPI) for the heterogeneous force. The resulting process represents a homogeneous force ratio attrition model, since this FPI is a single scalar value. The FPI, a single scalar measure of combat power, replaces the explicit representation of the heterogeneous force. The advantage of a firepower index-method lies in its ease of use and computational simplicity. Its disadvantage is the amount of subjectivity involved in determining the weight assigned to the different weapon systems. The experience and knowledge of experts assigning these weights might differ; consequently, there may be a discrepancy among the firepower values assigned by different experts. [Ref. 1]

The firepower index for a given Y (Red Force) unit can be represented as shown

$$FPI = \sum_{i=1}^{m} S_i * Y_i \tag{2.1}$$

where,

i = weapon systems index
m = number of different weapon systems
S_i = firepower score (value) for weapon system of type i
Y_i = number of weapons of type i

A similar approach is also used for the opposing X (Blue) force.

The force ratio (FR) is computed by dividing the attacker's (say Blue) firepower index by that of the defender (say Red). That is,

$$FR = \frac{FPI_{Blue}}{FPI_{Red}} \tag{2.2}$$

This gives a measure of relative combat power at the instant the values are calculated. [Ref. 1]

In large scale (i.e., division level and above) ground combat models, firepower indices are used as a surrogate for unit strength to determine engagement outcomes, assess casualties, and determine FEBA (Forward Edge of Battle Area) movement [Ref. 1]. However, subjectively determined firepower scores may vary widely for the same force. For this reason, the Force-Ratio approach has come under a fair amount of criticism [Ref. 10]. E.B. Vandiver, Director of the Center for Army Analysis (CAA), emphasized this point in his letter to A.W. Marshall, director of Net Assessment, Office of the Secretary of Defense, in April 1992:

> "Long experience teaches that so long as static scores exist, they will be badly misused. For this reason, I oppose their continued existence as a method of analysis." [Ref. 11]

Nevertheless, the Force Ratio approach is still used in some versions of current combat models such as TACWAR. [Ref. 12]

B. LANCHESTER-TYPE MODELS

A British engineer, F.W. Lanchester, is credited with the original introduction of Lanchester models. [Ref. 1] He developed these models in an attempt to quantify the effects of massing fires in modern warfare during World War I [Ref. 13].

A basic understanding of the underlying framework to which attrition rates are applied is necessary to the understanding of Lanchester-type [Ref. 1] attrition methodologies. A basic Lanchester-type [Ref. 1] analysis of modern warfare for the simple case of homogeneous forces will be illustrated in this chapter. This simple case is sufficient to illustrate all the salient points required for further understanding of the Bonder-Farrell and ATCAL applications of the Lanchester methodologies that are discussed below. The heterogeneous case that is implemented in this thesis is an extension of the homogeneous force case presented here.

Combat between two homogeneous forces, as conceptualized by the basic Lanchester-type [Ref. 1] analysis for modern warfare, assumes that the casualty rate of such a homogeneous force is directly proportional to the number of enemy firers. This is illustrated in Figure 1.

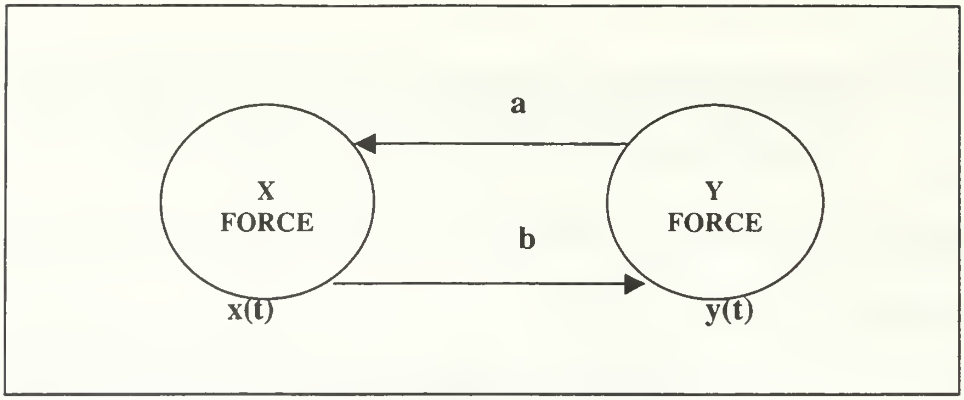

Figure 1. Basic Lanchester-type [Ref. 7] paradigm of modern warfare for homogeneous forces

In Figure I, X and Y are two hypothetical forces where x (t) and y (t) denote the number of X and Y firers respectively at time t. The rate at which a single typical Y firer kills X targets is "a" and it is called a Lanchester attrition-rate or a conditional kill rate. Similarly, "b" is the rate at which an X firer kills Y targets. For X > 0 and Y > 0, the basic Lanchester-type [Ref. 1] paradigm of modern warfare between homogeneous forces in Figure 1 can be expressed mathematically as the following:

$$\begin{aligned} \frac{dx}{dt} &= -a\,y \quad \text{with } x(0) = x_o \\ \frac{dy}{dt} &= -b\,x \quad \text{with } y(0) = y_o \end{aligned} \tag{2.3}$$

where, x_o and y_o represent initial force levels at time t=0. These equations must be turned off when x or y reaches zero. The values x and y are realizations of X and Y at time t.

According to the Bonder-Farrell attrition methodology [Ref. 12], the Lanchester attrition-rate-coefficient "a", can also described as the single-weapon-system-type kill rate, is computed by:

$$a = \frac{1}{E[T_{XY}]} \tag{2.4}$$

where $E[T_{XY}]$ denotes the expected time for an individual Y firer to kill an X target. The following section involves a description of the original problem faced by Bonder and a detailed description of $E\,[T_{XY}]$. It is followed by a discussion of the ATCAL approach in Section 2.

1. Bonder-Farrell Approach

The Bonder-Farrell attrition methodology for homogeneous forces of a single shooter, a tank, on either side is discussed here. This methodology involves a freestanding analytical model that generates attrition values from an analytical sub-model independent of any high-resolution simulation [Ref. 10]. The development of this methodology goes back to the original problem addressed by Bonder: the firing of a tank gun against a tank target. The U.S. Army's Armor School at Ft. Knox, KY, sponsored his original work. At the time of his research, gun fire control, i.e., laying and aiming, was achieved by using an optical telescope (gunsight) augmented by a reticle. In this type of fire control, the gun boresighted with the telescope was visually aimed at the target with each successive shot being re-aimed. If the gunner had to de-point the gun from the target to reload, then each shot was essentially a first shot. Luckily, this was mostly not the case except for very high angle shots where the gun would have to be almost leveled before the loading of the next round. The Armor School reduced the impact of this by developing special mechanisms for returning the gun to its original position, thus making each successive shot essentially a repeated shot. [Ref. 14]

The most significant factor in aiming the gun was whether or not the previous shot hit the target (assuming a kill did not take place). If it was a hit, then its impact point could usually be observed and the accuracy of the next round thereby increased. If the round was a miss, then its impact point was usually not observed (due to the flat trajectory), and thus the accuracy of the next shot could decrease. Some of these events are range-dependent factors (i.e., the acquisition and flight times, and probabilities of hit and kill) that objectively represent the possible sequence of events in an actual one-on-one duel between two opposing tanks. [Ref. 14]

Bonder expressed the attrition-rate coefficient as the reciprocal of the expected time between casualties. He related the attrition-rate coefficient to time dependent factors that are adjusted to fit the flow of the battle. The basic equation developed by Bonder to obtain the expected time to kill a target, E[T], can be computed by summing all the component event times resulting in a combat kill [Ref. 12]. Bonder calculated the expected time to kill a target with the following set of equations:

$$E[T] = t_a + t_1 - t_h + A_1 + A_2 * [A_3 + p(h \backslash h) - p_1] \quad (2.5)$$

$$A_1 = (t_h + t_f) / p(k \backslash h) \quad (2.6)$$

$$A_2 = (t_m + t_f) / p(h \backslash m) \quad (2.7)$$

$$A_3 = (1 - p(h \backslash h)) / p(k \backslash h) \quad (2.8)$$

where:

t_a = time to acquire the target,
t_1 = time to fire the first round after acquisition,
t_h = time to fire a succeeding round given the previous one was a hit,
t_m = time to fire a succeeding round given the previous one was a miss,
t_f = time of flight of the round to the target,

$p(k\backslash h)$ = probability of kill given a hit,
$p(h\backslash m)$ = probability of hit of a succeeding round given the previous one missed,
$p(h\backslash h)$ = probability of hit of a succeeding round given the previous one hit,
p_1 = probability of a first round hit. [Ref. 14]

All times mentioned above in the sets of equations (2.5) - (2.8) are expected values. For example, only the mean time to acquire a target, or the mean time to fire the first round after acquisition etc. are used in this model. Furthermore, there is perfect information concerning whether or not the target is killed. The model and information provided in these equations are for a single shooter, but display the salient features of the approach. However, for application to practical problems, one needs to consider combat between heterogeneous forces, a topic that is beyond the scope of this thesis. A similar Bonder-Farrell approach is also used for heterogeneous forces.

2. ATCAL Approach

The ATCAL approach involves two phases. ATCAL Phase I estimates Lanchester attrition-rate coefficients from the output of a high-resolution Monte-Carlo combat simulation. In ATCAL Phase II, the set of Lanchester-type [Ref. 1] equations and the Lanchester attrition-rate coefficients obtained from Phase I are used to simulate aggregate force attrition. [Ref. 3]

The ATCAL methodology forms the basis for force-on-force casualty assessment in direct-fire and area-fire engagements in large-scale computer-based ground combat models. It is used by the CAA to extrapolate results of the high-resolution COSAGE simulation runs into the aggregated CEM model. [Ref. 3]

There is essentially no documentation of the theoretical basis of the ATCAL methodology used by the CAA. Furthermore, existing ATCAL documentation does not even mention that the ATCAL methodology is based on a Lanchester-type [Ref. 1] combat attrition model. [Ref. 10]

The next sections provide a thorough documentation of the underlying Lanchester-type [Ref. 1] model for ATCAL that was adapted by James G. Taylor [Ref. 15] and [Ref. 8]. First, the homogeneous and heterogeneous force models are given. Next, the derivation of the ATCAL assessment equations from the underlying Lanchester-type [Ref. 1] model is discussed to clearly establish the theoretical basis of ATCAL.

a) Homogeneous-Force Model

According to Clark, the basic direct-fire, Lanchester-type model for the case of two homogeneous forces has the following form of nonlinear equations [Ref. 8].

For $x > 0$ and $y > 0$,

$$\begin{aligned} \frac{dx}{dt} &= -\alpha(1 - p^{x})y \quad \text{with } x(0) = x_0 \\ \frac{dy}{dt} &= -\beta(1 - q^{y})x \quad \text{with } y(0) = y_0 \end{aligned} \tag{2.9}$$

The x and y variables in the equations in (2.9) are the sizes of the X and Y forces. x_0 and y_0 denote the size of X and Y forces at time $t = 0$. The constants $\alpha > 0$ and $\beta > 0$ are the conditional kill rates, while the constants p, q are the probabilities of target non-availability. That is,

α = Conditional kill rate, also known as the Lanchester attrition-rate coefficient
(i.e., Rate at Which a Y Firer Kills X Targets)

(2.10)

p = Prob [Typical Y Firer Does Not Have a Particular
X Target Available to Engage]

and similarly for q and β. It is usually more convenient to speak in terms of target availability defined, for example, as follows

$$A = 1 - p \tag{2.11}$$

A = Prob [Typical Y Firer Has a Particular X target
Available to Engage]

and

$$B = 1 - q \tag{2.12}$$

B = Prob [Typical X Firer Has a Particular Y Target
Available to Engage]

This results in the following form for equation (2.9):

$$\begin{aligned} \frac{dx}{dt} &= -\alpha\{(1-(1-A)^{x}\}y \quad \text{with } x(0) = x_o \\ \frac{dy}{dt} &= -\beta\{(1-(1-B)^{y}\}x \quad \text{with } y(0) = y_o \end{aligned} \tag{2.13}$$

It should be noted that attrition modeled by the above differential equations must be turned off when the number of targets has been reduced to zero to prevent negative force levels from occurring. [Ref. 8]

b) Heterogeneous-Force Model

A heterogeneous force model should be used in situations where an X force that is composed of m different types of weapon systems is opposed by a Y force

composed of n different types of weapon systems. In this scenario, the underlying Lanchester-type model has the following form of (m+n) nonlinear Clark equations [Ref. 8]:

For $x_i > 0$ and for $y_j > 0$,

$$\frac{dx_i}{dt} = -\sum_{j=1}^{n} \alpha_{ij} \{(1 - p_{ij}^{x_i}) \prod_{k=i+1}^{m} p_{kj}^{x_k}\} y_j \quad \text{for } i = 1, \dots, m,$$

$$\frac{dy_j}{dt} = -\sum_{i=1}^{m} \beta_{ji} \{(1 - q_{ji}^{y_j}) \prod_{k=j+1}^{n} q_{ki}^{y_k}\} x_i \quad \text{for } j = 1, \dots, n, \tag{2.14}$$

Here i = 1 denotes Y's lowest priority X target type and m denotes the highest priority X target type. Similarly, j = 1 denotes X's lowest priority Y target type and n denotes X's highest priority Y target type. The index k represents the higher priority target types for i and j firers. The following initial conditions apply:

x_{io}= initial X force size at t=0 for i = 1,…,m,

and (2.15)

y_{jo}= initial Y force size at t=0 for j = 1,…,n.

Equations in (2.14) are turned off when x_i or y_j reach zero.

The constants α_{ij} and $\beta_{ji} > 0$ are the conditional kill rates. The constants p_{ij}, $q_{ji} > 0$ are the probabilities of target non-availability for all pertinent values of the indices i and j. The constants p_{kj} and q_{ki} are the probabilities of target non-availability of higher priority target types for all pertinent values of the indices i and j. That is,

p_{kj} = Prob [Typical Y_j Firer Does Not Have a Particular Higher Priority X_k Target Available to Engage]

p_{ij} = Prob [Typical Y_j Firer Does Not Have a Particular X_i Target Available to Engage] (2.16)

α_{ij} = Conditional kill rate, also known as the Lanchester attrition-rate coefficient (i.e., Rate at Which a Y_j Firer Kills Acquired X_i Targets),

and similarly for q_{ki}, q_{ji} and β_{ji}. In terms of target availability, the underlying Lanchester-type [Ref. 1] model of combat between two-heterogeneous forces takes the following form:

For $x_i > 0$ and $y_j > 0$,

$$\frac{dx_i}{dt} = -\sum_{j=1}^{n} \alpha_{ij} \{(1-(1-A_{ij})^{x_i}) \prod_{k=i+1}^{m} (1-A_{kj})^{x_k}\} y_j \quad \text{for } i = 1,\ldots,m$$
$$\frac{dy_j}{dt} = -\sum_{i=1}^{m} \beta_{ji} \{(1-(1-B_{ji})^{y_j}) \prod_{k=j+1}^{n} (1-B_{ki})^{y_k}\} x_i \quad \text{for } j = 1,\ldots,n \tag{2.17}$$

All target types are arranged in order of increasing target priority, i.e., higher priority target types have higher index values. The definitions of the probabilities of target availability A_{ij}, B_{ji}, A_{kj} and B_{ki}, are included. [Ref. 8]

A_{ij} = Prob [Typical Y_j Firer Has a Particular X_i Target Available to Engage]

B_{ji} = Prob [Typical X_i Firer Has a Particular Y_j Target Available to Engage]

(2.18)

A_{kj} = Prob [Typical Y_j Firer Has a Particular Higher Priority X_k Target Available to Engage]

B_{ki} = Prob [Typical X_i Firer Has a Particular Higher Priority Y_k Target Available to Engage].

Equations in (2.17) are turned off when x_i or y_j reach zero.

The development of the underlying Lanchester-type [Ref. 1] model for ATCAL for homogeneous and heterogeneous forces has been provided. The next section explains the development of a Lanchester-type model for ATCAL and establishes its theoretical basis.

c) *The Development of a Lanchester-Type Model for ATCAL*

The derivation of ATCAL assessment equations from the underlying Lanchester-type model depends on a transformation of the Lanchester-type [Ref. 1] differential equations by an averaging operator. This operator was first used, but never documented by Dr. Alan Johnsrud when he developed the ATCAL methodology for CAA in 1983. [Ref.15]

Consider that the average of a function x(t), $\bar{x}$, over time is given by the following:

$$\bar{x} = \frac{1}{t}\int_0^t x(s)\,ds = \mathrm{Ave}[x] \tag{2.19}$$

Furthermore,

$x = x(t)$ denotes the X force level at time t

x_o denotes the initial X force level

$\Delta x(t) = x_o - x(t)$ denotes X force casualties at time t

Application of the averaging operator to a Lanchester-type [Ref. 1] model results in an equation for the force levels. Consider the following linear Lanchester-type [Ref. 1] model explained previously in equation (2.3). For $x > 0$ and $y > 0$,

$$\begin{aligned} \frac{dx}{dt} &= -a\,y \quad \text{with } x(0) = x_o \\ \frac{dy}{dt} &= -b\,x \quad \text{with } y(0) = y_o \end{aligned} \tag{2.20}$$

The constants "a" and "b" in the above equations represent the Lanchester attrition-rate coefficients as explained in equation (2.3).

Applying the averaging operator to dx / dt results in

$$\text{Ave}\left[\frac{dx}{dt}\right] = \frac{1}{t}\int_0^t \frac{dx}{ds}\,ds = \frac{1}{t}\{\,x(t) - x_o\,\} \tag{2.21}$$

or

$$\text{Ave}\left[\frac{dx}{dt}\right] = -\frac{\{x_o - x(t)\}}{t} = -\frac{\Delta x(t)}{t} \tag{2.22}$$

Applying the operator to the right-hand side of the first equation in (2.20) yields

$$\text{Ave}[ay] = a\,\text{Ave}[y] = -a\,\bar{y}\,. \tag{2.23}$$

Finally, a similar application of the averaging operator to the second equation in (2.20) results in the following equations with initial force levels of $x > 0$ and $y > 0$:

$$\begin{aligned} \Delta x(t) &= a\,t\,\bar{y}(t) \\ \Delta y(t) &= b\,t\,\bar{x}(t) \end{aligned} \tag{2.24}$$

Two important points need to be emphasized here. First, for a linear Lanchester-type model, such as equation (2.20) above, the averaging operator is exact and linear in the average force levels. Second, the equations contain twice the number of unknowns as the original Lanchester-type [Ref. 1] model. [Ref. 15]

This second point implies that, if there are m unknown force levels in the linear Lanchester-type [Ref. 1] model, then there will be m such unknown casualty figures (i.e., values of $\Delta x(t)$ and $\Delta y(t)$) and also m unknown values for the average force levels in the equations. Thus the model is underdetermined and m additional constraints must be added to equations (2.24) in order to be able to solve them explicitly. Johnsrud supplied these additional conditions by assuming that all force levels undergo an exponential decay. [Ref. 3] He assumes that the following condition, for the X force level, holds,

$$\bar{x}(t) = \frac{(-\Delta x(t))}{\ln\left(1 - \frac{\Delta x(t)}{x_o}\right)}, \tag{2.25}$$

where,

$\bar{x}$ is the time average of force level x(t) and is defined by (2.19)

Δx is the number of X casualties during the assessment period of length t ($x_o - x(t)$)

x_o is the initial number of X force units.

Taylor has shown [Ref. 16] that if equation (2.25) holds, then x(t) must be an exponential function, either decreasing or increasing over time. As a result, for the case of homogeneous forces, ATCAL solves the following system of four nonlinear

equations in four unknowns ($\bar{x}, \bar{y}$, Δx, and Δy) in (2.26) by successive substitution to determine the numbers of casualties during each assessment period.

The discussion above demonstrates the derivation of basic ATCAL assessment equations from an underlying Lanchester-type [Ref. 1] model and thus establishes the theoretical basis of ATCAL.

$$\begin{aligned} \Delta x &= \alpha\, t(1-p^{\bar{x}})\,\bar{y} \\ \Delta y &= \beta\, t(1-q^{\bar{y}})\,\bar{x} \\ \bar{x} &= \frac{(-\Delta x)}{\ln\left(1-\frac{\Delta x}{x_o}\right)}, \\ \bar{y} &= \frac{(-\Delta y)}{\ln\left(1-\frac{\Delta y}{y_o}\right)}, \end{aligned} \qquad (2.26)$$

ATCAL is a Lanchester-type [Ref. 1] model whose theoretical foundations were established by Gordon M. Clark in his Ph.D. thesis in 1969 [Ref. 7]. Clark used [Ref. 7] MLE of parameter values from the time-stamped output data of the high-resolution Monte-Carlo combat simulation in his thesis. CAA did not use his thesis when ATCAL was being developed in 1982. For this reason, CAA did not adopt a MLE approach for the estimation of model parameters in ATCAL. In chapters IV and V, this thesis demonstrates the application of a MLE approach originally envisioned by Clark.

This chapter has shown the development of homogeneous and heterogeneous force ATCAL equations. The development of a Lanchester-type [Ref. 1] model for

ATCAL and the establishment of its theoretical basis has also been achieved. The next step in this research is to obtain the high-resolution simulation output data in order to estimate the values of the conditional kill rates, α_{ij} and β_{ji}, and the target availabilities, A_{ij} and B_{ji}, in the ATCAL equations that were described in this chapter.

III. JANUS COMBAT MODEL

In Chapter III, a combat scenario is created in JANUS in order to obtain the required time-stamped attrition and target acquisition data for the estimation of the parameters mentioned above. Parameter estimation involves the application of COMAN MLE methodology, which is the topic of Chapter IV. A detailed description of the target acquisition process in JANUS is essential before explaining the COMAN maximum likelihood estimation approach to obtain target availabilities from the high-resolution simulation output.

A. OVERVIEW

1. Background of Development

JANUS, named after the two-faced Roman God who was the guardian of the gates of Rome and the patron of beginnings and endings, exists in several versions. The initial version was developed as a nuclear effects modeling simulation by the Lawrence Livermore National Laboratory at the University of California and called JANUS (Livermore) or JANUS (L) [Ref. 9]. It was delivered to the United States Army Training and Doctrine Command (TRADOC) at White Sands Missile Range, New Mexico in January 1983 [Ref. 17] where it was adapted for use in tactical training. Several versions of JANUS have been developed and used since 1983 for both training and analysis. Version 6.88 (Unix Model) of JANUS is used in this research effort.

2. Characteristics

The JANUS combat model is an interactive, six-sided, closed, stochastic ground combat simulation with a graphical user interface [Ref. 9]. Interactive refers to the military analyst's responsibility for the control of position and movement of forces and decisions on what to do during crucial situations of the simulated combat. The direct fire engagements are scripted. However, the user can plan artillery missions or move his forces when JANUS is run in the interactive mode. In the automatic mode, the user sets up the battle and allows it to run on its own. Six-sided refers to the possibility of as many as six opposing units controlled by six separate sets of players. This feature allows a simulation of coalition warfare. Closed means that the disposition of opposing sides is unknown to the players in control of the other sides with the exception of the information provided by those friendly units in contact with the opposing units. That is, only those enemy units detected by friendly forces appear on the friendly side's screen. Stochastic means that the results of events in the simulated battle, like direct fire engagements, are determined according to the laws of probability. The same results may or may not occur if the same scenario is repeated. The user has the option of setting a random number seed between 1 and 100 for each JANUS run. If the same random number seed is used with the same scenario in a future run, results of the original experiment will be repeated when JANUS is run in the automatic mode. Ground combat refers to the fact that the principle focus of the model is on maneuver warfare and artillery units. [Ref. 9]

In addition, JANUS also models weather and its effects, day and night visibility, engineering support, minefield employment and breaching, rotary and fixed-wing aircraft,

resupply operations and chemical warfare [Ref. 9]. The terrain representation in JANUS realistically affects visibility and movement. The combat model uses digitized terrain data developed by the Defense Mapping Agency (DMA), in order to provide military users with familiar terrain features such as contour lines. Other terrain features unique to the terrain selected such as roads, vegetation, rivers and urban areas can be created by the user and pasted onto the terrain file provided by DMA. These terrain features replicate their real world characteristics such as visibility through vegetation or trafficability on a certain type of road created. All the features of the simulation described above provide the user with the flexibility to accurately represent a multitude of combat scenarios and thus makes JANUS the primary high-resolution combat simulation used by the United States Army to model brigade sized (and below) combat operations. [Ref. 9]

B. TARGET ACQUISITION PROCESS

1. Background and Purpose

The target acquisition process in JANUS is based on the Night Vision Electronic Sensors Directorate (NVESD) target acquisition methodology, referred to as the ACQUIRE methodology. This was adopted by the U.S. Army in 1993 [Ref. 18]. The ACQUIRE methodology basically uses the same methodology as the Army's Center for Night Vision and Electro-Optic devices (CNVEO) search model [Ref. 19]. JANUS used the CNVEO model until the adoption of the ACQUIRE methodology in 1993 [Ref. 18]. The integration of the ACQUIRE methodology into JANUS has been accomplished through data input. A table of P-infinity values, which indicate the fraction of a large

population of observers that would eventually detect a given target after a very long time, is loaded during the initialization process [Ref. 20]. In addition, the way in which P-infinity values for the set of observers and targets are determined was also slightly modified [Ref. 20]. However, these results did not significantly alter JANUS results [Ref. 20]. Some features and parameters of the search and detection process remain subjects of debate and modifications are occasionally made when appropriate [Ref. 20].

The ACQUIRE model produces a single-glimpse probability of acquisition by considering the number of resolvable cycles across the target and requires input from the target platform, atmospheric objects and the sensor object in the simulation conducting the search. Since JANUS is interactive, the requirements for rapid response time strongly influence the implementation of the ACQUIRE model. Pre-processing of simulation objects before the computationally heavy line-of-sight (LOS) calculations reduces the number of computations required during the execution of the scenarios. Additionally, filters reduce the computationally expensive trips through the search algorithms. [Ref. 19]

The search and detection process is implemented in three stages in JANUS. In the first stage, initialization, the necessary parameters are set up in order to partition the search as a function of the units on each side and the two time constraints (target list update cycle time and detection cycle time) that are input by the analyst. Sensor performance level characteristics for each sensor type are also initialized for every observer-target pair in the scenario. During the scenario runs, units, which meet the acquisition criteria, are placed on a potential target list. This constitutes the second stage of the search process. In the third stage, the probability of detection of those targets on

the potential target list is calculated for specific observers. Random draws are then made to determine which targets are actually detected. [Ref. 19]

The following sections explain the parts of the ACQUIRE model that are applicable to this thesis in terms of the simulation of target acquisition and discrimination in JANUS. These topics will be discussed in order to make them understandable to readers who are not familiar with the computer simulation techniques used by JANUS or the details of the search and detection theory. Interested readers may refer to [Ref. 18], [Ref.19], [Ref. 21] and the references contained therein for more detailed information on target acquisition in JANUS.

2. Initialization of the Search and Detection Process

JANUS is an interactive simulation and the search and LOS processes consume most of the computational resources. Thus, a trade off exists between the frequency of search updates and the responsiveness of an analyst's interactions. The analyst is allowed to specify a target list update cycle time in JANUS Screen I (Appendix A) that is longer than the detection cycle time. This allows the units to be considered less often for the filtering processes and saves some computation at the expense of a somewhat less accurate list of potentially detectable targets. The choices of the detection and target list update cycle times influence the JANUS' performance in a given scenario. Shorter cycle times will generally allow a better representation of the search process. However, they will result in increased run times and longer interactive response times for a given game. [Ref. 19]

a) Filters

The detection process in JANUS uses a first stage of filtering units in order to avoid the computationally costly search and LOS calculations for as many units as possible. Most of the logic and bookkeeping for filtering units is contained in the DSTLOS subroutine. A unit is filtered out if it is destroyed, inoperable due to chemical warfare agents, or if it is a passenger and cannot detect targets. Units must go through filtering before they can be added to the potential target list. [Ref. 19]

b) Potential Target List

Targets that pass the filters mentioned above and whose cycles are resolved in the field-of-view (FOV) of the sensor, meet the criterion for eventual detection. They are placed on the observer's potential target list with an associated value reflecting its ease of detection. There can only be a total of ten targets on a given observer's list. If the target list is full, the least detectable target in the list is replaced by a more detectable one. Those targets detected during previous cycles have priority and are always placed on the potential target list. [Ref. 19]

3. Explanation of Resolvable Cycles

The probability of target detection in the ACQUIRE model is dependent upon the concept of the number of resolvable cycles across a target's presented minimum dimension for each target-sensor device pair and the time the target spends in the sensor's field of view. Visualize a pattern of stripes equal in width at the target's location as in Figure 2. Assume that they alternate in color and the contrast between the colors is the

same as the contrast between the target and the surrounding background. Also let the length of target's minimum-presented dimension be represented by the length of the pattern in Figure 2. One can easily distinguish with the naked eye the number of black and white lines in Figure 2. [Ref. 21]

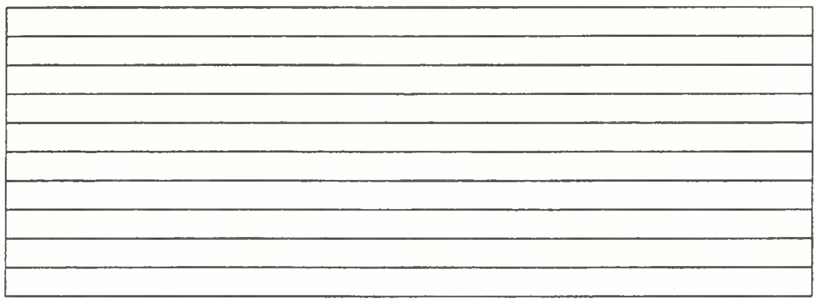

Figure 2. Target's minimum presented dimension at the target's location represented as a pattern of stripes equal in width. Figure taken from Ref [22].

Decreasing the distance between the black and white lines to a minimum width until the eye cannot distinguish between them is shown in Figure 3.

This minimum width is analogous to the wavelength of one cycle in the ACQUIRE model and defines the sensitivity or capability of the eye. The same thing is true for the sensors used in JANUS simulation. It should also be noted that this cycle wavelength represented by the minimum width varies with the contrast between the target and the background. Using this width, the number of resolvable cycles can be defined as the number of pairs of stripes contained within a distance equal to the target's minimum presented dimension. [Ref. 22]

Figure 3. Decreasing the distance between the black and white lines to a minimum is shown here. This minimum distance is analogous to the wavelength of one cycle in the ACQUIRE model. Figure taken from [Ref. 22].

JANUS uses the following parts of the ACQUIRE model to calculate detection probabilities and also to compute the probability of distinguishing a target at a higher level of resolution given the number of cycles resolved:

a. The attenuation in the atmosphere of the target's signature along the line of sight between the target and the sensor is computed. [Ref .21]

b. Using the target's signature computed in the first step, the number of resolvable cycles (N) that the sensor can resolve across the target's minimum-presented dimension is calculated. [Ref. 21]

c. Using the number of resolvable cycles, N, a decision is made to determine if the target can be detected. If so, the time and the level of resolution of target detection is also determined. [Ref. 21]

Each of the three steps above is discussed in detail below.

a) Attenuation in the Atmosphere

The target's signature travels through the atmosphere before reaching the sensor. As illustrated in Figure 4, this signature is affected by normal atmospheric conditions and also by an obscurant called the "large-area" smoke cloud in JANUS [Ref. 21].

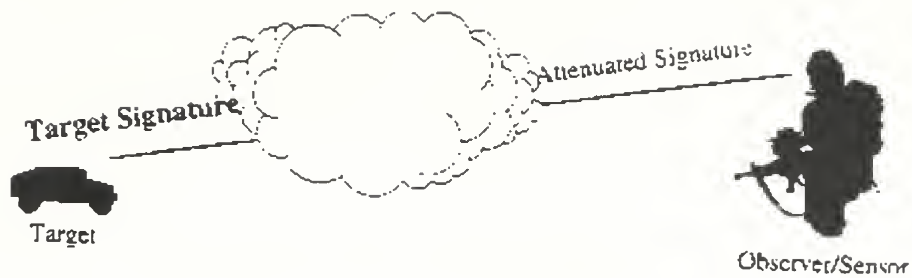

Figure 4. A large-area smoke cloud blocking the line-of-sight between the target and the observer's sensor. Figure taken from [Ref. 22].

The attenuated target signature at the sensor can be represented as follows:

$$Ss = St * T1 * T2 \tag{3.0}$$

where,

Ss = Attenuated target signature at the sensor

St = Signature at target

T1 = The attenuation due to transmission of the normal atmosphere

T2 = The attenuation of transmission caused by any large-area smoke cloud along the LOS between the target and the sensor.

T1 and T2 have values between 0 and 1. That is, if both of them are one, then there is no attenuation. If either of them is zero, the target's signature is not transmitted to the sensor, and the sensor cannot detect the target. For optical sensors, St represents the optical contrast of the target. The optical contrast is part of the weather data in the master database in JANUS. The normal atmospheric transmission for optical sensors can be described as follows. [Ref. 21]:

$$T1 = \frac{1}{1 + SGR * (e^{(\alpha * R)} - 1)} \tag{3.1}$$

where,

R = range between target and sensor

α = The extinction coefficient of the atmosphere, which is part of the weather data stored in the master database

SGR = The sky-to-ground brightness ratio, which is also part of the weather data and stored in the master database.

For thermal sensors, St is the absolute value of the average temperature difference between the target and its surrounding background. The atmospheric transmission for thermal sensors can be written as [Ref. 21]:

$$T1 = e^{(-\alpha * R)} \tag{3.2}$$

where, α and R are defined as above.

b) Resolvable Cycles

The cycles per milliradian (CMR) is the measure of sensor sensitivity. It is the number of resolvable cycles that the sensor can distinguish as mentioned previously. The CMR can be expressed as a function of the target signature at the sensor using Ss from equation (3.0). That is,

$$CMR = f\,(Ss). \tag{3.3}$$

However, there is no closed form solution for the above equation. Instead, JANUS uses a mean resolvable contrast (MRC) curve for optical sensors and a mean resolvable temperature (MRT) curve for thermal sensors. JANUS uses the interpolation of the MRC and MRT data to obtain CMR values [Ref. 21]. After the CMR value is obtained from

the MRC and MRT curves through interpolation, the number of resolvable cycles, N, is then calculated by the following equation [Ref. 21]:

$$N = CMR * (TDIM / RANGE) \quad (3.4)$$

where,

TDIM = Target's minimum presented dimension

RANGE = Range between the target and the sensor

Finally, after the number of resolvable cycles for a particular target-sensor combination is determined, the probability of detection of the target can be determined using the method described in the following section.

4. The Probability of Target Detection

JANUS only considers those targets that are on the potential target list for detection. The probability of detection of the target is a function of the number of resolvable cycles, calculated in equation (3.4), across the target's minimum-presented dimension [Ref. 22].T

The following definitions are necessary for use in determining the probability of detection:

P_∞ = The probability that the target will eventually be detected given sufficient (infinite) time [Ref. 21].

P(t) = The probability of the target being detected during a time interval of t, given that the target can, in fact, eventually be detected while it is still in the sensor's field of view [Ref. 21].

N = The number of resolvable cycles across the target's critical dimension [Ref. 22].

N50 = The median number of resolvable cycles needed for eventual detection [Ref. 21].

$$R = (N / N50) \text{ [Ref. 22]} \tag{3.5}$$

$$E = 2.7 + 0.7 * (N / N50) \text{ [Ref. 22]} \tag{3.6}$$

Finally, the P_∞ value is given by:

$$P_\infty = R^{\left(\frac{E}{(1+R^E)}\right)} \tag{3.7}$$

JANUS generates a uniform random number on the interval 0 to 1 and compares it to the P_∞ value computed for that target-unit and sensor-unit pair. If the P_∞ value is smaller than the random number, then the target is not considered for detection. If the P_∞ value is greater than the random number, the target passes the P_∞ test and is considered for detection. However, it is not detected yet. Once the above P_∞ test for detection is passed, JANUS uses the exponential distribution to model when the target is actually detected. [Ref. 22] The rate of detection for the exponential distributed random detection time is estimated in the following way using the P_∞ value calculated previously:

$$\lambda = \begin{cases} \dfrac{P_\infty}{3.4}, & \text{for } \dfrac{N}{N50} \leq 2 \\ \dfrac{N}{(6.8\,(N50))}, & \text{for } \dfrac{N}{N50} > 2 \end{cases} \tag{3.8}$$

The probability of detection during the time interval t, the amount of time the target has spent in the sensor's field of view, is computed by entering the λ value into the standard exponential cumulative distribution function. That is,

$$P_{\text{Detection}}(t) = 1 - e^{-\lambda t} \tag{3.9}$$

JANUS compares the probability of detection value computed as explained above to a new random number on the interval 0 to 1 during each target detection cycle [Ref. 22]. This is repeated until the target is either detected during that detection cycle, or is no longer available for detection by the observer (i.e., the LOS is broken, target is destroyed or moves out of range, or the observer is killed, etc.). [Ref. 21]

C. LEVELS OF TARGET DISCRIMINATION

The level of resolution at which a detection event occurs can be varied by altering the value of N50 parameter used to generate the P_∞ distribution [Ref. 18]. JANUS uses four levels of target discrimination (resolution):

1. Detection: This is the ability of an observer to distinguish an object of military interest that is foreign to the background in its field of view (FOV), e.g., distinguishing a vehicle from a bush [Ref. 18].

2. Aimpoint: It simply refers to the ability of the observer to distinguish a target by its class, e.g., a tracked vehicle vs. a helicopter or a wheeled vehicle [Ref. 18]. The observer can thus establish an aimpoint on the object, which has been determined to be of military interest [Ref. 21].

3. Recognition: This refers to the ability of the observer to categorize targets discriminated at aimpoint within a given class, e.g., recognizing a tank versus an armored personnel carrier (APC) in the tracked vehicle class [Ref. 18].

4. Identification: This is the ability of the observer to distinguish between specific recognized target models, e.g., a T-72 tank versus a T-80 [Ref. 18].

The corresponding N50 values used in JANUS for these levels of target discrimination are listed in Table 1 below [Ref. 21]:

Target Discrimination Level	Criteria (cycles / milliradian)
Detection	1.0
Aimpoint	2.0
Recognition	3.5
Identification	6.4

Table 1. N50 values used in determining the levels of target discrimination in JANUS.

Detection indicates that something foreign to the background is present in the observer's field of view. Aimpoint, recognition, and identification represent different levels of target acquisition. A value of 2.0 resolvable cycles is used if the target acquisition is simulated at aimpoint level. It should be noted that a collection of random draws generated by the P_∞ distribution for a given value of N50 can also be scaled to a collection of random draws generated by other values of N50. For example, $P_\infty(1.0)$ is the distribution with N50=1.0 and $P_\infty(3.5)$ represents the P_∞ distribution with N50=3.5. Thus, multiplying each draw from $P_\infty(1.0)$ by a scalar of 3.5 is equivalent to a collection of random draws from $P_\infty(3.5)$. A random draw from the $P_\infty(1.0)$, the "detection threshold", is stored for every target-unit/sensor-unit during JANUS' initialization stage. This random draw is scaled up by a scalar of 2.0 when the target is evaluated for acquisition at aimpoint level. The observer unit can acquire the target if the scaled-up draw is greater than or equal to 2.0 resolvable cycles. The original detection threshold value is scaled by 3.5 for recognition and by 6.4 for identification to determine if the target can be discriminated at higher levels of discrimination than aimpoint during every

detection cycle after the target has been acquired. The target discrimination level is then used as an input parameter to the direct-fire engagement algorithms used in JANUS. [Ref. 21]

D. FIRING CRITERIA

The direct-fire engagement algorithms use firing criteria similar to the levels of target discrimination used in target acquisition. The reader must take care not to confuse the similarity in terminology between the firing criteria and the levels of target discrimination. The conditions under which units with direct-fire weapons engage targets are called the firing criteria. JANUS combat model uses three firing criteria:

1. **Aimpoint:** Lowest acquisition level is aimpoint. It means that units may fire at detected targets without regard for their type. This is the least restrictive firing criterion [Ref. 9].

2. **Recognition:** This is more restrictive that aimpoint and requires knowing the type of unit (e.g., tank vs. APC in the tracked vehicle class). [Ref. 9].

3. **Identification:** This is the most restrictive firing criterion. It requires identification of the specific system e.g. T-72, M1 or Bradley Fighting Vehicle). This is the most restrictive firing criterion. [Ref. 9]

The analyst can change the firing criterion on JANUS Screen IV (Appendix B). It is also important to note that the firing criterion applies only to engagement areas. The firing criterion outside of engagement areas is identification and cannot be changed by the user. Engagement areas are similar to sectors of responsibility that are established by the orders that state the conditions under which specific units are supposed to fire their weapons in a tactical situation.

The analyst can create, enable and disable the engagement areas on individual workstations. Enabling engagement areas means that the firing criteria apply to all of the engagement areas created for that workstation. If the engagement areas are disabled on a workstation, then the entire map area on that workstation becomes the engagement area. If the engagement areas are enabled and no engagement areas are defined, then the entire map area on that workstation is considered to be outside of the engagement area and the firing criteria is identification. [Ref. 9]

The definitions of firing criteria and the levels of target discrimination explained above will be utilized for the COMAN maximum likelihood estimation approach in Chapter IV to obtain target availability for the heterogeneous force mixes. The next section explains the JANUS scenario that was used in this thesis to obtain time-stamped attrition and acquisition data necessary for the estimation of ATCAL Phase I parameters.

E. JANUS SCENARIO

1. Environment and Force Structure

The environment selected for the JANUS scenario was Hunter Liggett, California (HL). HL is primarily a lightly wooded terrain with mountains. However, the engagement takes place at a relatively flat part of the terrain with small hills. U.S. and Russian-style armored and mechanized units, along with their supporting artillery elements, were selected as the unit types. The scenario involves two U.S.-equipped mechanized task-force battalions (Blue) in attack against two Russian-equipped armored companies (Red). The Blue weapon systems used were 56 M1A1 tanks, 56 M2IFV

Infantry Fighting vehicles, and eight Combat Engineer Vehicles (CEV). Attacking Blue units were supported by 12 M109A3 155mm self-propelled howitzers (two batteries). The Red side consisted of 12 T-72 tanks, eight T-80 tanks, eight BMPs, and eight BMP-2s in defense. The Red side was supported by six 2S3 self-propelled artillery units. The Blue side had a total of 132 weapon systems, while the red side had a total of 38 weapon systems, resulting in more than a 3:1 ratio in favor of the attacker. The scenario was designed to examine the capabilities of direct fire weapon systems used such as the M1A1, T-72, T-80, M2IFV, BMP, and BMP-2s.

The detection and firing capabilities of entities differ in JANUS depending on the posture of the unit. The engagement type selected (an attacking force against a defending force) provides results from entities with different postures, i.e. moving and stationary.

2. Scenario Execution

In JANUS, the artillery fire can be either planned during the execution of a scenario or can be pre-planned before the scenario execution starts. In order to ensure the randomness of each different run of the scenario, a no-man-in-the-loop design was necessary. Thus, only preplanned fires were used in the execution of the scenario.

All Red units were placed on relatively defensible terrain with prepared defensive positions oriented in three different directions. The initial disposition of forces is shown in Figure 5. This figure represents one run of the scenario created in JANUS. The file number for this run of the scenario is 15028. The first three numbers, 150, represent the scenario number. The last two numbers, 28, represent the run number for scenario 150.

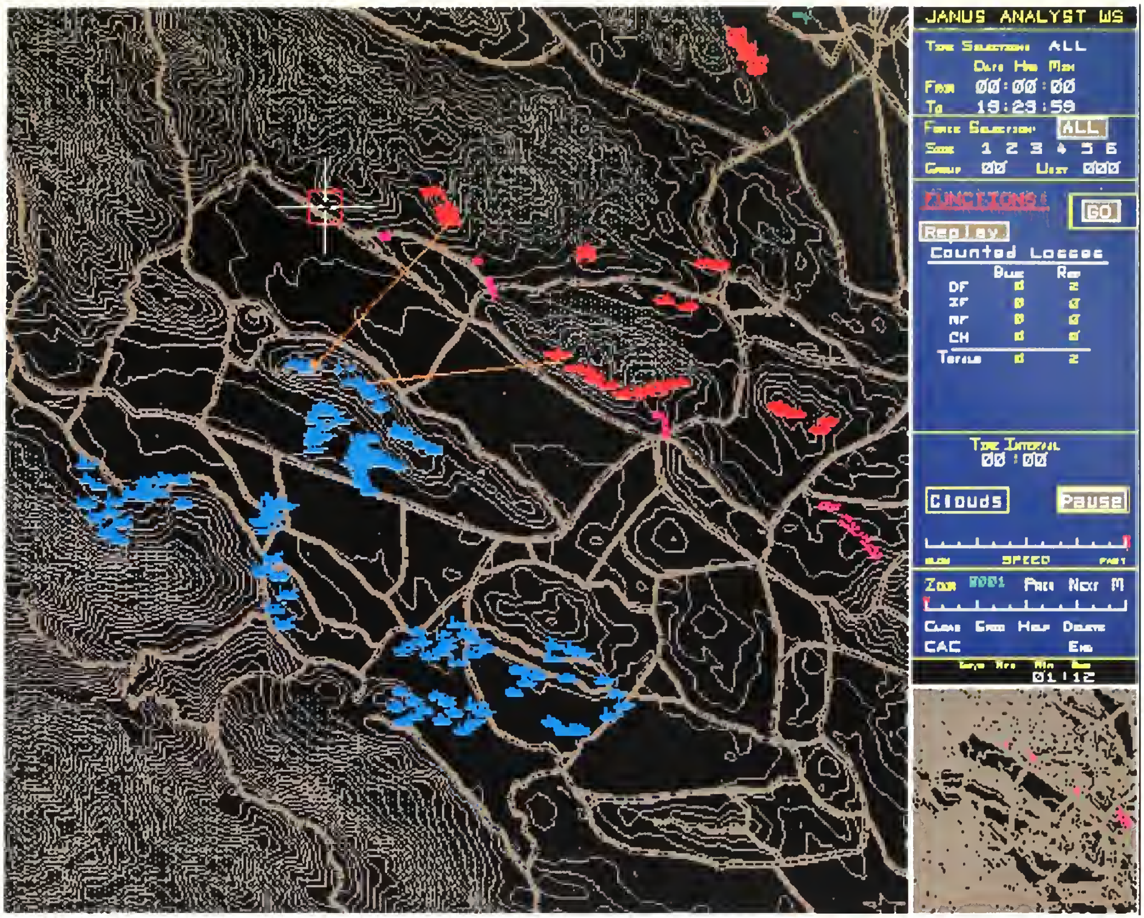

Figure 5. The initial disposition of forces at the start of the battle for scenario 15028. The orange lines indicate the direct-fire engagements taking place between the Blue and the Red sides. The purple figures represent the defensive minefields emplaced by the defending Red units. Brown lines are the contour lines of the Hunter Liggett terrain. Yellow and green colors represent the vegetation.

The Blue units attack from the Southeast and the Northwest in two different directions in an attempt to encircle the Red forces. Once the attack from these two directions reaches the intended limits of advance, Blue launches a third attack crossing the Forward Edge of Battle Area (FEBA) in the middle. The disposition of forces at approximately half way through the battle is shown in Figure 6.

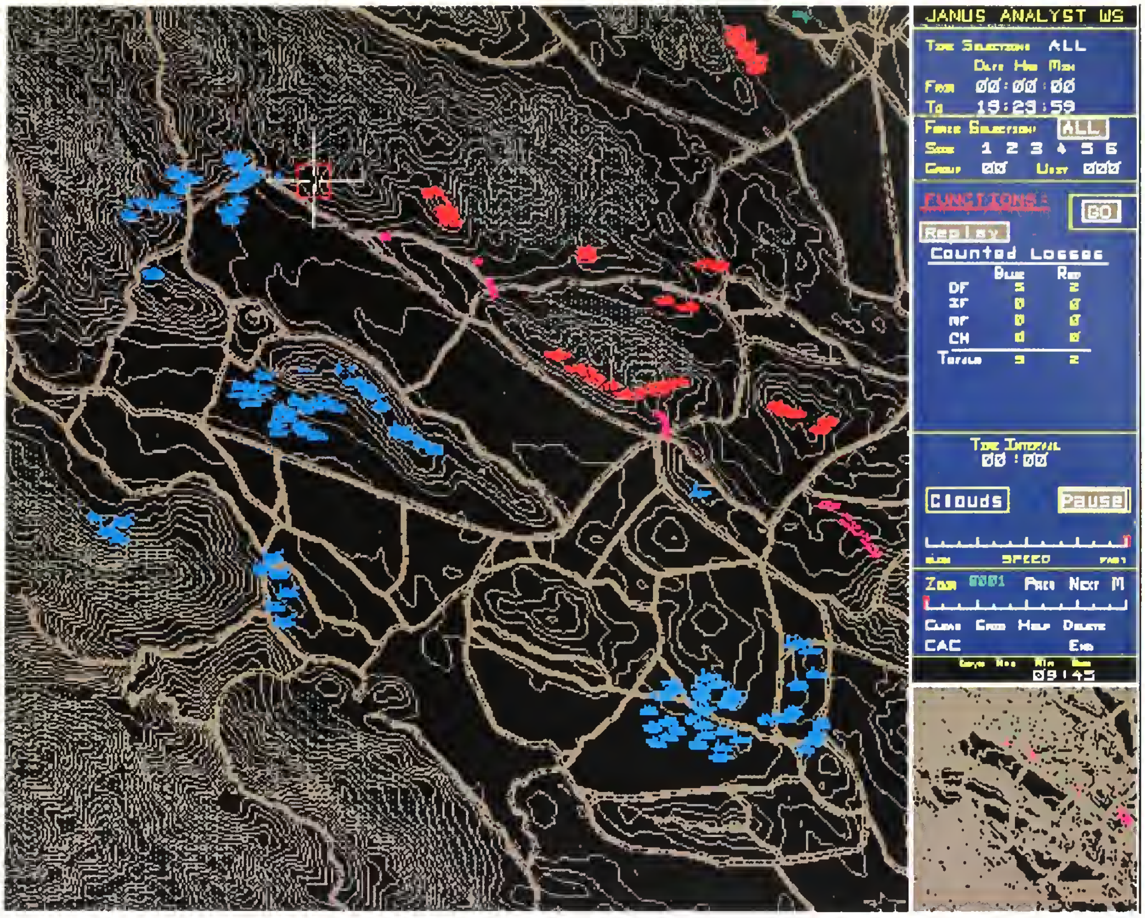

Figure 6. The disposition of forces at approximately half way through the battle for scenario 15028. Blue's attack from two different directions is almost in position to support a third attack from the middle of FEBA.

The Blue forces eventually succeed in capturing the Red defensive positions. Both sides suffer heavy casualties. However, the artillery units on both sides do not suffer any casualties. This is due to the fact that no counter-battery artillery fire was planned during the scenario run and that no direct fire weapons on either side reached the other side's artillery units' positions. The disposition of forces just before the end of the battle is shown in Figure 7.

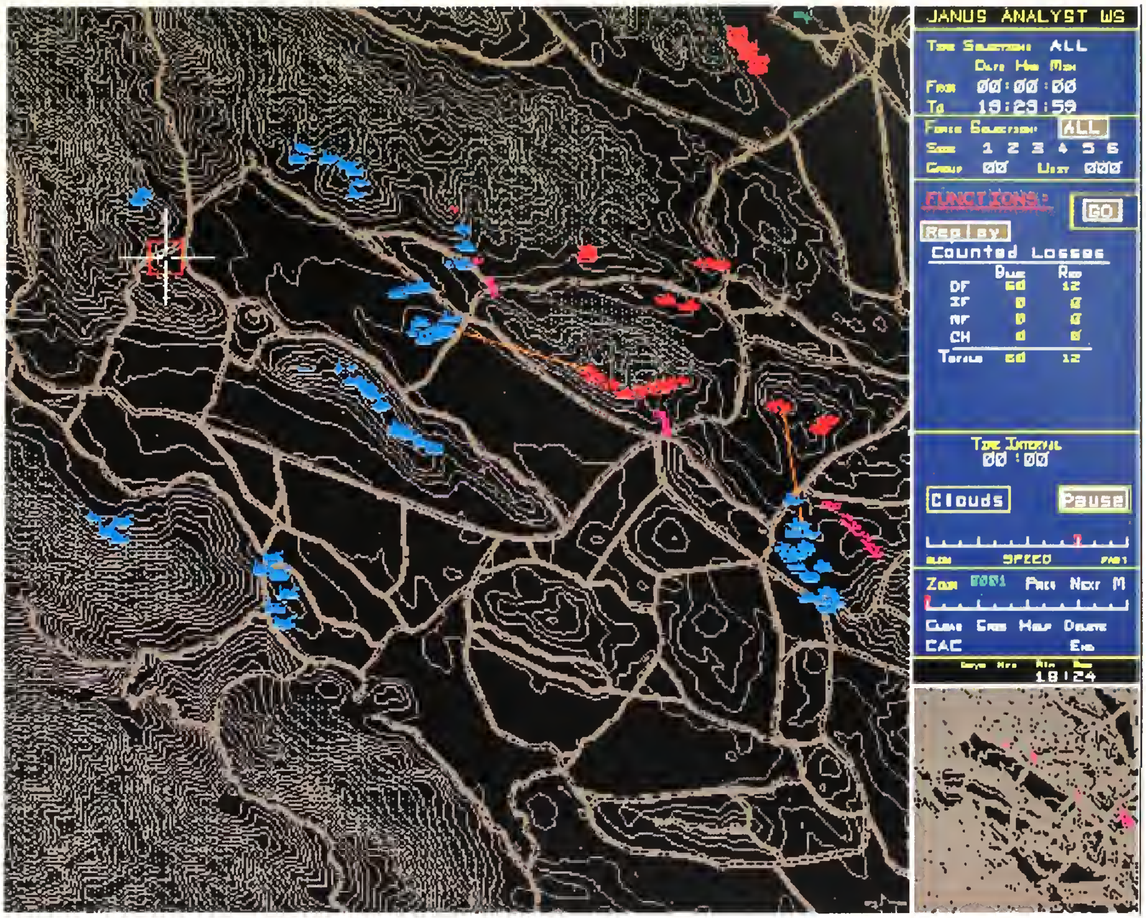

Figure 7. The disposition of forces right before the end of battle for scenario 15028. Blue and Red artillery units suffer no casualties.

A total of 20 runs of the same scenario were made with different random number seeds. Each run shows slightly different results. Ten of these runs were made using the recognition firing criterion for both sides. The other ten runs were made using the more restrictive identification firing criterion for both sides. The random number seeds used in scenario runs are provided in Table 2 below.

Run Number	Recognition	Identification	Run Number
15028	97	97	15040
15029	83	83	15041
15030	11	11	15042
15031	52	52	15043
15032	16	16	15044
15033	1	1	15045
15034	3	3	15046
15035	69	69	15047
15036	46	46	15048
15037	31	31	15049

Table 2. The random number seeds used when running the JANUS scenario for different firing criteria

Twenty detection files, twenty direct-fire kill files and a systems file that lists the 170 weapon systems used in the simulation were obtained. A subset of the contents of these files are used. The results of these files are used for the time period during which a combined total of 57 kills occur for reasons that will be explained in Chapter IV. The remaining data is discarded after a combined total of 57 kills occur for Blue and Red forces. Chapter IV explains the COMAN maximum likelihood estimation approach for homogeneous and heterogeneous forces. The use of a continuous-time, three-state Markov chain model to obtain target availability parameters in COMAN is also a topic of discussion in Chapter IV.

IV. THE COMAN MAXIMUM LIKELIHOOD ESTIMATION APPROACH

The maximum likelihood estimates (MLE) of the parameters (α, β, A and B) used in the nonlinear Clark equations (2.9) will be derived here.

The Clark equations underlie the ATCAL methodology as explained in Chapter II. Parameter estimation is the key for obtaining ATCAL input from the output of a high resolution Monte-Carlo simulation of ground warfare, such as JANUS.

The COMAN approach develops a series of maximum likelihood estimates for Lanchester attrition-rate coefficients by using input parameters from a high resolution simulation with various force mixes, tactical situations, weapon characteristics and different terrain selections. These estimates are balanced with the probability of an opposing target being undetected (i.e., the target availability of the opposing side) and also with the prioritization of targets for the firing unit. These estimates are then used to determine the attrition of forces within each time step in an aggregated simulation model. The results of the COMAN model can also be utilized to extrapolate the results of the high-resolution simulation to predict the combat outcome from larger number of force mixes than were not explicitly simulated [Ref. 7]. There are other statistical methods besides MLE to obtain point estimates of this type (such as the method of moments, Bayesian estimation and the method of least squares). However, MLE is the only approach that has had a significant application in combat models [Ref. 7]. The use of a MLE approach for model parameters makes the COMAN attrition rates asymptotically

unbiased and normally distributed with minimum variance because the method of maximum likelihood has the property that the estimators it produces are always functions of sufficient statistics [Ref. 23].

A. COMAN MAXIMUM LIKELIHOOD FUNCTION FOR HOMOGENEOUS FORCES

The derivation of the maximum likelihood function will be given in detail for the homogeneous force case. The results are then extended to heterogeneous forces without a detailed derivation in section B of this chapter.

The following system of equations are the same as the system of equations (2.13). The maximum likelihood estimation of conditional kill rates, α and β, and target availabilities, A and B, of equation (4.1) below for the case of force-on-force direct fire combat between two homogeneous forces is presented:

$$\begin{aligned} \frac{dx}{dt} &= -\alpha\{(1-(1-A)^{x}\}y \quad \text{with } x(0)=x_0 \\ \frac{dy}{dt} &= -\beta\{(1-(1-B)^{y}\}x \quad \text{with } y(0)=y_0 \end{aligned} \tag{4.1}$$

where the constants $\alpha>0$ and $\beta>0$ are the attrition rates,

where,

A = Prob [Typical Y Firer Has a Particular X target
Available to Engage]

(4.2)

B = Prob [Typical X Firer Has a Particular Y Target
Available to Engage]

1. Notation for the COMAN MLE

The following definitions are necessary:

α = conditional kill rate of the X force target by the Y force firer

β = conditional kill rate of the Y force target by the X force firer

M (t) = number of Y combatants remaining at time t with realization m

N (t) = number of Y combatants remaining at time t with realization n

$m_o = M(0)$ is the initial number of X combatants at t=0

$n_o = N(0)$ is the initial number of Y combatants at t=0

Assume data obtained from JANUS can be represented by the Lanchester-type [Ref. 1] equation (4.1). A continuous-time Markov chain, which has a discrete state space (corresponding to the integer numbers of combatants on each side) and allows a random occurrence of casualties, is an appropriate stochastic model of the Lanchester-type [Ref. 1] equation presented above. [Ref. 24] This stochastic model has the following Forward-Kolmogorov equations [Ref. 25] corresponding to the deterministic Lanchester-type analogue presented in equation (4.1).

For $0 < m < m_o$ and $0 < n < n_o$,

$$\frac{dP}{dt}(t,m,n) = A(m+1,n)\,P(t,m+1,n) + B(t,m,n+1)\,P(t,m,n+1) - \{A(m,n)+B(m,n)\}\,P(t,m,n) \tag{4.3}$$

where,

$$P\,(t, m, n) = \text{Prob}\begin{bmatrix} M(t) = m \mid M(0) = m_o \\ N(t) = n \mid N(0) = n_o \end{bmatrix} \tag{4.4}$$

$$A\,(m, n) = \alpha\,\{1 - (1 - A)^m\}n \tag{4.5}$$

$$B\,(m, n) = \beta\,\{1 - (1 - B)^n\}m \tag{4.6}$$

The modified output from the JANUS Scenario 150 gives the times at which casualties occur (and also the type of each casualty) during the simulation. This stochastic simulation is run until a both sides experience a combined total of K casualties. The total run time of the simulation is a random variable that is denoted here as T_K (with realization t_k). In addition, let C_k^X and C_k^Y (with realizations c_k^x and c_k^y) be defined by the following [Ref. 12]:

$$C_k^X = \begin{cases} 1 & \text{if the } k^{th} \text{ casualty is an X combatant} \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

$$C_k^Y = \begin{cases} 1 & \text{if the } k^{th} \text{ casualty is a Y combatant} \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

Realizations c_k^x and c_k^y are important since $c_k^x \cdot c_k^y = 0$ and $c_k^x + c_k^y = 1$. Let C_T^X and C_T^Y denote the total number of casualties for each side. That is,

$$C_T^X = \sum_{k=1}^{K} c_k^x \quad \text{and} \quad C_T^Y = \sum_{k=1}^{K} c_k^y \tag{4.9}$$

with, $C_T^X + C_T^Y = K$.

In addition, let m_k and n_k represent m (t_k) and n (t_k) respectively where,

m_k = size of X force after the k^{th} casualty

n_k = size of Y force after the k^{th} casualty

In other words, there are m_k X-force and n_k Y-force survivors during the interval $[t_k, t_{k+1})$ for $k = 0, 1, \ldots, K$.

2. Maximum Likelihood Estimation Steps:

Using the times and types of casualty data $(t_1 \ldots t_k, c_1^x \ldots c_k^x,\ c_1^y \ldots c_k^y)$ mentioned above, the MLEs of the parameters A, B, α, β for equation (4.1) will be developed. Taylor states that for the continuous-time Markov-chain model presented above, the three-step process below [Ref. 25] may be used to determine the MLEs.

(1): Find the probability density function (P.D.F) for the time until an X (respectively a Y) casualty occurs.

(2): Construct the likelihood functions for X and Y forces.

(3): Determine the values of the parameters that maximize the likelihood functions.

These three steps will be discussed in detail below:

Step (1): Assume the probability density functions f_{s_x} and respectively f_{s_y} of the time between casualties for X (respectively Y), can be written as:

$$f_{s_x}(s) = r_{xy}\, e^{-(r_{xy} + r_{yx})s} \tag{4.10}$$

$$f_{s_y}(s) = r_{yx}\, e^{-(r_{xy} + r_{yx})s} \tag{4.11}$$

where,

S_x = random variables denoting the time between any two consecutive X casualties

S_y = random variables denoting the time between any two consecutive Y casualties

and the following variables represent respectively the rates at which an X firer kills Y targets and the rate at which a Y firer kills X targets.

$$\begin{aligned} r_{yx} &= r_{yx}(\alpha,\beta,m,n)=\beta\{1-(1-B)^n\}m \\ r_{xy} &= r_{xy}(\alpha,\beta,m,n)=\alpha\{1-(1-A)^m\}n \end{aligned} \tag{4.12}$$

Note that A and B are independent of n and m.

Step (2): Using the memoryless property of the Markov process allows the likelihood function to be simply the product of the likelihoods for each of the independent kill-time events. That is, the occurrence of the k^{th} casualty at t_k. (k-1) represents the the time when (k-1)st casualty occurs. r_{xy} (k-1) represents the rate at which a Y firer kills X targets at the time of the (k-1)st casualty. The likelihood function is given by the following:

$$L(\alpha,\beta,A,B)=\prod_{k=1}^{K}(r_{xy}(k-1))^{c_k^x}\ (r_{yx}(k-1))^{c_k^y}\ e^{-(r_{xy}(k-1)+r_{yx}(k-1))(t_k-t_{k-1})} \tag{4.13}$$

Taking the natural logarithm of the likelihood function yields:

$$\ln L(\alpha,\beta,A,B)=\sum_{k=1}^{K} c_k^x \ln r_{xy}(k-1)+\sum_{k=1}^{K} c_k^y \ln r_{yx}(k-1)-\sum_{k=1}^{K}\{r_{xy}(k-1)+r_{yx}(k-1)\}(t_k-t_{k-1}) \tag{4.14}$$

Step (3): Find the values of parameters that maximize the likelihood functions requires taking the first partial derivative with respect to α yields,

$$\frac{\partial}{\partial\alpha}\ln L=\frac{c_T^x}{\alpha}-\sum_{k=1}^{K}\frac{r_{xy}(k-1)}{\alpha}(t_k-t_{k-1}) \tag{4.15}$$

Substituting for r_{xy} and setting this first partial derivative equal to zero gives an estimate

for α,

$$\hat{\alpha}=\frac{c_T^x}{\sum_{k=1}^{K}\{1-(1-\hat{A})^{m_{k-1}}\}n_{k-1}(t_k-t_{k-1})} \tag{4.16}$$

Similarly, $\hat{\beta}$ is the estimate for β given by the following equation:

$$\hat{\beta}=\frac{c_T^y}{\sum_{k=1}^{K}\{1-(1-\hat{B})^{n_{k-1}}\}m_{k-1}(t_k-t_{k-1})} \tag{4.17}$$

The variables n_{k-1} and m_{k-1} represent the number of Y and X firers remaining respectively at the time of the (k-1)st casualty. The equations (4.16) and (4.17) for estimates of the conditional kill rates α and β can only be solved if the estimates of target availability A and B can be calculated. Taking the natural logarithm of the likelihood function with respect to A yields

$$\frac{\partial}{\partial A}\ln L=\sum_{k=1}^{K}c_k^x\left\{\frac{m_{k-1}(1-A)^{m_{k-1}-1}}{1-(1-A)^{m_{k-1}}}\right\}-\alpha\sum_{k=1}^{K}m_{k-1}(1-A)^{m_{k-1}-1}n_{k-1}(t_k-t_{k-1}) \tag{4.18}$$

By setting this equation equal to zero and using the estimate of α as given by equation (4.16), the two-dimensional optimization problem can be reduced to a one-dimensional one. The estimate of A, the target availability for Y firers, is that value that maximizes the likelihood function in terms of target availabilities for X-force targets. This point must also satisfy the following nonlinear equation:

$$\hat{f(A)}= \sum_{k=1}^{K} c_k^x \left\{ \frac{m_{k-1}(1-\hat{A})^{m_{k-1}-1}}{1-(1-\hat{A})^{m_{k-1}}} \right\} - c_T^x \frac{\left[\sum_{k=1}^{K} m_{k-1}(1-\hat{A})^{m_{k-1}-1} n_{k-1}(t_k - t_{k-1}) \right]}{\left[\sum_{k=1}^{K} \left\{ 1-(1-\hat{A})^{m_{k-1}} \right\} n_{k-1}(t_k - t_{k-1}) \right]} = 0 \tag{4.19}$$

The sign of the second derivative can also be investigated to determine if the point that maximizes equation (4.19) is a maximum. If the sign of the second derivative is negative, this point must be a maximum. On the other hand, if the sign of the second derivative is positive, then this point must be a minimum. A similar equation for the availability of Y-forces is the following [Ref. 24]:

$$\hat{g(B)}= \sum_{k=1}^{K} c_k^y \left\{ \frac{n_{k-1}(1-\hat{B})^{n_{k-1}-1}}{1-(1-\hat{B})^{n_{k-1}}} \right\} - c_T^y \frac{\left[\sum_{k=1}^{K} m_{k-1}(1-\hat{B})^{n_{k-1}-1} m_{k-1}(t_k - t_{k-1}) \right]}{\left[\sum_{k=1}^{K} \left\{ 1-(1-\hat{B})^{n_{k-1}} \right\} m_{k-1}(t_k - t_{k-1}) \right]} = 0 \tag{4.20}$$

$(t_k - t_{k-1})$ represents the time between kills of the k^{th} and the (k-1)st casulaty. The target availability calculated from equations (4.19) and (4.20) above can be substituted into equations (4.16) and (4.17) to estimate α and β, the attrition rates for X and Y forces respectively.

MLEs for the continuous-time Markov-chain combat model corresponding to the homogeneous-force nonlinear Clark equations [Ref. 7] have been developed in this section. These ideas are now extended to the case of heterogeneous forces.

B. COMAN MAXIMUM LIKELIHOOD FUNCTION FOR THE HETEROGENEOUS FORCES

The maximum-likelihood estimation of the target availability is much more complicated for heterogeneous forces than it is for homogeneous forces. However, the development of maximum-likelihood estimates for the conditional kill rates for heterogeneous forces is still fairly straight forward, provided that one knows the values for target availability. In the first section of this chapter, the determination of target availability necessary for the computation of conditional kill rates is explained. A three-state Markov chain model that is used to obtain target availability values is also explained in this section. Target availability values are substituted into the MLEs for conditional kill rates in equations (4.21) and (4.22) below.

The variance of the ATCAL Phase I MLEs of conditional kill rates have historically been a concern for the analysts [Ref. 7], [Ref. 17]. Thus a discussion of the variance of MLEs is the topic of the second section in this chapter. The last section in this part of Chapter IV discusses the establishment of confidence intervals for the MLEs.

The MLEs for the heterogeneous force conditional kill rates are the following equations (4.21) and (4.22). Similar definitions mean different things for each equation. The words "firer" and "targets" are written in italics in the definitions below to prevent confusion.

$$\hat{\alpha}_{ij} = \frac{C_T^{X_i Y_j}}{\sum_{k=1}^{K} \{1-(1-\hat{A}_{ij})^{m^i_{(k-1)}}\} \prod_{\ell \in I_{ij}} (1-\hat{A}_{\ell j})^{m^{\ell}_{(k-1)}} \, n^j_{(k-1)} (t_k - t_{(k-1)})} \quad \text{for } i = 1,..m \tag{4.21}$$

where, for equation (4.21)

$\hat{\alpha}_{ij}$ = estimate of the conditional kill rate of X *targets* by Y *firers*

$\hat{A}_{ij}$ is the availability of X *targets* to Y *firers*

$c_T^{X_i Y_j}$ = total number of X_i killed by Y_j

$m^i_{(k-1)}$ = number of X_i *targets* remaining just before the k^{th} casualty combined for both sides

$n^j_{(k-1)}$ = number of Y_j *firers* remaining just before the k^{th} casualty combined for both sides

I_{ij} is the set of all higher priority X_i target types for Y_j firers

$\ell = \ell \in I_{ij} \mid \ell$ is integer and X target type ℓ has a higher priority than target type i for a Y_j firer

$m^{\ell}_{(k-1)}$ = number of X *targets* with higher priority than target type i remaining just before the k^{th} casualty

$(1-\hat{A}_{\ell j})$ = probability of not having a higher priority X_i target type available for a Y_j *firer*

$(t_k - t_{(k-1)})$ = time between kills on both sides

and for equation (4.22),

$$\hat{\beta}_{ji} = \frac{c_T^{Y_j X_i}}{\sum_{k=1}^{K} \{1-(1-\hat{B}_{ji})^{n^j_{(k-1)}}\} \prod_{\ell \in J_{ji}} (1-\hat{B}_{\ell i})^{n^{\ell}_{(k-1)}} m^i_{(k-1)} (t_k - t_{(k-1)})} \quad \text{for } j=1,..n \qquad (4.22)$$

$\hat{\beta}_{ji}$ = estimate of the conditional kill rate of Y_j *targets* by X_i *firers*

$\hat{B}_{ji}$ is the availability of Y_j *targets* to X_i *firers*

$c_T^{Y_jX_i}$ = total number of Y_j killed by X_i

$m^i_{(k-1)}$ = number of X_i *firers* remaining just before the k^{th} casualty for both sides

$n^j_{(k-1)}$ = number of Y_j *targets* remaining just before the k^{th} casualty for both sides

J_{ji} is the set of all higher priority Y_j target types for X_i *firers*

$\ell = \ell \in J_{ji} \mid \ell$ is integer and Y target type ℓ has a higher priority than target type j for an X_i firer

$n^{\ell}_{(k-1)}$ = number of Y targets with higher priority than target type j remaining just before the k^{th} casualty

$(1-\hat{B}_{\ell i})$ = probability of not having any higher priority Y_j target type for an X_i firer

$(t_k - t_{(k-1)})$ = time between kills on both sides

A 1990 Naval Postgraduate School thesis [Ref. 17] investigated the estimation of the heterogeneous force conditional kill rates. The thesis used the following equations:

$$\hat{\alpha}_{ij} = \frac{c_T^{X_iY_j}}{\sum_{k=1}^{K} n^j_{(k-1)}(t^j_k - t^j_{(k-1)})} \quad \text{for } i = 1,..m \tag{4.23}$$

$$\hat{\beta}_{ji} = \frac{c_T^{Y_jX_i}}{\sum_{k=1}^{K} m^i_{(k-1)}(t^i_k - t^i_{(k-1)})} \quad \text{for } j = 1,..n \tag{4.24}$$

All of the variables are defined as previously except for the times between kills. This study used the times between kills of firers and not the times between any kills as discussed in equations (4.21) and (4.22). The current research project used [Ref. 17] as a reference and attempted to apply times between kills of firers in equations (4.23) and (4.24). It was discovered that the author of [Ref. 17] assumed the target availability values were 1.0 for each firer-target combination in equations (4.23) and (4.24), and also assumed there were no priorities between the targets to allocate fires. This current research does not make these assumptions. The methodology in [Ref. 17] cannot be applied when these assumptions do not hold.

This approach in [Ref. 17] was tried as part of this thesis research. The JANUS scenario explained in Chapter III was run twenty times to obtain a time series data of kills. This data was manipulated by a JAVA program written to determine the times between kills of firers. However, the results obtained could not be substituted into the COMAN equations (4.21) and (4.22) described this chapter. The methodology in [Ref. 17] is flawed. Instead, the times between all kill events, and not the times between kills of firers as done in [Ref. 17], were used. In addition, target priority and target availability information were used in the current research as defined in equations (4.21) and (4.22). This model described above by equations (4.21) and (4.22) requires a more detailed simulation output than the thesis done in 1990 or the model considered for the homogeneous case discussed previously in this chapter. This detailed output from JANUS is used to obtain the target availability values in equations (4.21) and (4.22).

1. Target Availability for the Heterogeneous Force

Target availability will be determined by first estimating the target-acquisition parameters for this new model. Conditional kill rates will be estimated in a way similar to that mentioned above for homogeneous forces. Prof. Gordon M. Clark suggested considering a more detailed stochastic model to obtain the estimates for target availabilities. This model is a finite-state Markov chain that allows target availability to be expressed in terms of target-acquisition parameters [Ref. 7]. Figure 8 illustrates the Markov chain model that was suggested for determining the target availabilities of the heterogeneous case.

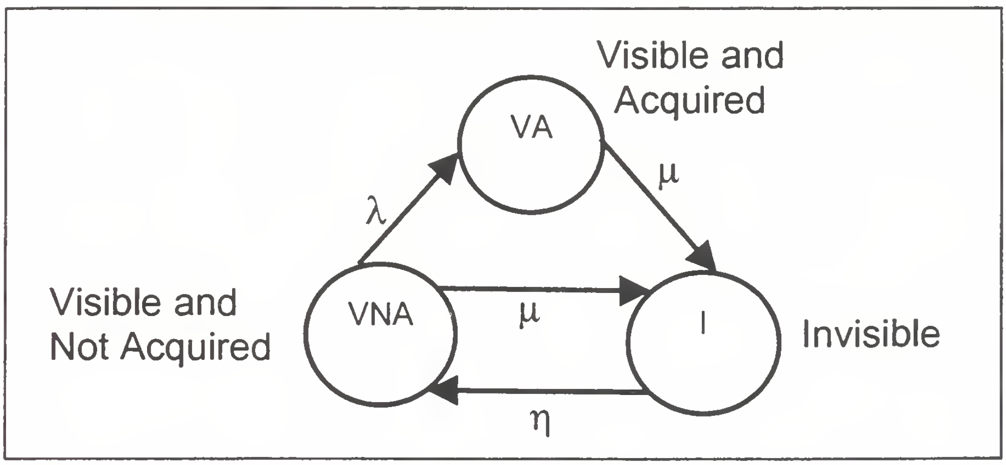

Figure 8. The original three-state Markov chain model of non-firing target acquisition suggested by Gordon Clark.

JANUS, the high-resolution combat model used for this research, allows a transition in target behavior from the Invisible state to the Visible and Acquired state. Therefore, the three-state Markov chain model depicted above must be modified and a different expression for target availability must be developed before it can be used to

obtain target availability of heterogeneous forces. The modified three-state Markov chain model that accommodates this change in target behavior is illustrated in Figure 9:

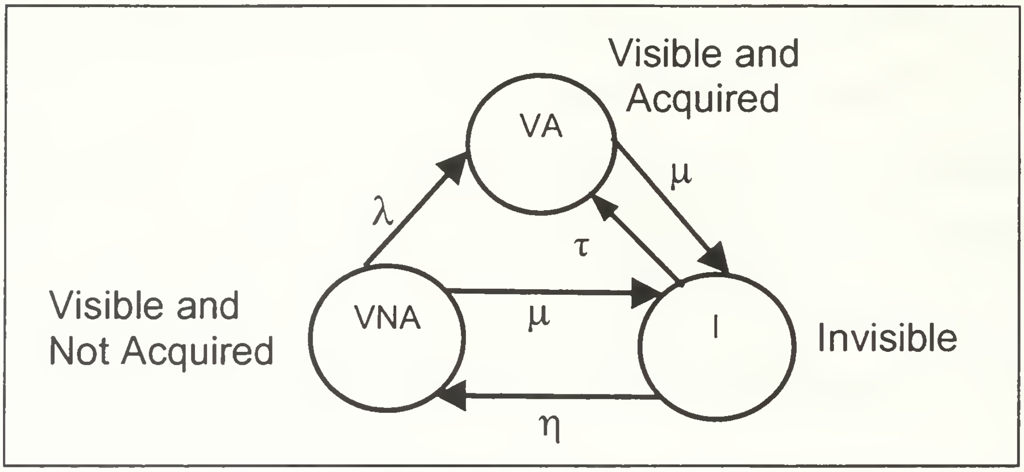

Figure 9. The modified version of the original three-state Markov chain model of non-firing target acquisition suggested by Gordon Clark. During the research effort, the author of this thesis discovered that JANUS allows a transition from the Invisible state to the Visible and Acquired state.

The Markov chain in Figure 9 models the non-firing acquisition of targets. The loss of LOS due to a kill event is not included in this model. Only detection events are modeled to obtain the number targets available for each weapon system to fire at. The states in this model represent the target's state as seen by the observer. The Invisible (I) state means that the target has not been seen by the observer unit, or the LOS between the observer and the target has been lost. The Visible and Not Acquired (VNA) state means that the target has been seen by the observer at a certain level of discrimination, depending on the firing criterion selected, but has not yet been placed on the observer's potential target list as explained in Chapter III. The Visible and Acquired (VA) state means that the target has been observed at the highest level of discrimination, depending on the firing criterion selected, and is placed on the observer's potential target list.

JANUS outputs the times of detections and firing events for every weapon system in its standard post processing files. However, it does not provide the times when a firer loses LOS to a target. These data exist and are used internally in JANUS' routines, but are not included as the standard output in the JANUS post processing files. These data were extracted by using a TRADOC Analysis Center (TRAC)/Monterey modification of the JANUS program that is written in FORTRAN. This modification adds additional output routines into the JANUS program. These source code changes are required to force JANUS to output these times when LOS is lost. Data was output to a text file, rather than to the standard JANUS post processing files. These text files are referred to as the modified detection files in this research. This allowed easier manipulation of the data by the JAVA program, which computed the rates and the steady state solution of the Markov chain model depicted in Figure 9. An executable version of the modified JANUS program that outputs these text files is available from TRAC/Monterey.

There are three different firing criteria in JANUS as explained in Chapter III. Therefore, the next step in this chapter is to explain how the levels of target discrimination and the selected firing criterion map into the Markov chain model of non-firing acquisition depicted in Figure 9. Table 3 displays a sample modified JANUS detection file obtained as a result of the process described above by running the scenario explained in Chapter III and used for determining target availability.

Observer Unit ID Number	Target Unit ID Number	Level of Target Discrimination	Time of Detection
4	202	0	0.025
4	258	0	0.025
4	324	0	0.025
5	242	0	0.025
8	170	0	0.025
9	180	0	0.025
9	175	0	0.025
9	324	2	0.025
10	208	0	0.025
16	307	0	0.025
17	227	0	0.025
17	180	0	0.025
4	258	3	0.05
8	170	2	0.05
9	324	4	0.05
10	208	2	0.05
11	307	1	0.05
12	180	1	0.05
12	242	2	0.05
15	307	3	0.05
17	186	3	0.05
15	307	4	0.075
17	227	4	0.075
10	227	2	0.075
4	258	4	0.075

Table 3. A sample modified detection file output by JANUS. Detection files are used to determine target availability.

The levels of target discrimination displayed in column three of Table 3 above correspond to the levels of target discrimination explained in Chapter III. These numbers represent the levels of target discrimination displayed in Table 4:

Levels of Target Discrimination	
Numerical Value	**Meaning of the Numerical Value**
0	Target unit is not detected by the observer unit
1	Target unit is detected at aimpoint level of discrimination
2	Target unit is detected at recognition level of discrimination
3	Target unit is detected at identification level of discrimination
4	The LOS between the observer and the target is lost

Table 4. Explanation of the numerical output of JANUS corresponding to the levels of target discrimination.

The JANUS detection file does not discriminate between the loss of LOS due to kill events and other causes, i.e., an entity becoming invisible due to large area smoke clouds or the effects of terrain features. Thus, the transition between the VA and the I states is an over-censored process. That is, there would be more transitions from VA to I than from VNA to I. Therefore, this transition is not used in steady state calculations. Instead, only the transition rate from the VNA state to the I state will be used in the steady-state equations explained below in order to get the target availabilities. There are three firing criteria in JANUS as explained in Chapter III. The identification and the recognition firing criteria are relevant for the three-state Markov chain model in Figure 9. Depending on the firing criterion selected, the levels of target discrimination output in the modified detection file in Table 3 correspond to different states in the model in Figure 9. This is important in determining the transition rates of the model. If the firing criterion is recognition, then the states in the Markov chain model correspond to the levels of discrimination depicted in Figure 10 below:

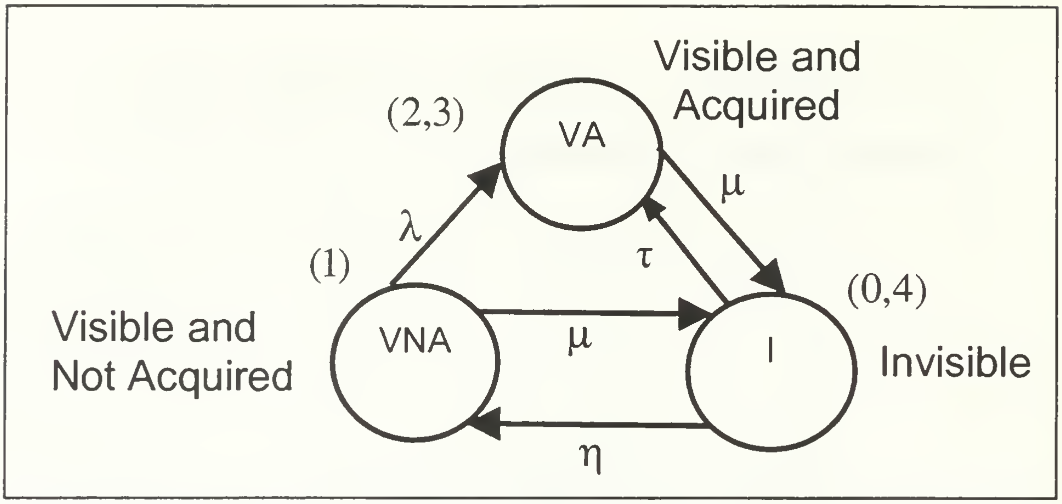

Figure 10. The levels of target discrimination output in the modified JANUS detection file corresponding to the states in the Markov chain model of target acquisition if the recognition firing criterion is used. Targets discriminated as a result of levels of discrimination two or three are acquired. They will be fired upon automatically as a result of the direct-fire algorithms.

If the firing criterion is identification, then the states in the Markov chain model correspond to the levels of discrimination depicted in Figure 11 below:

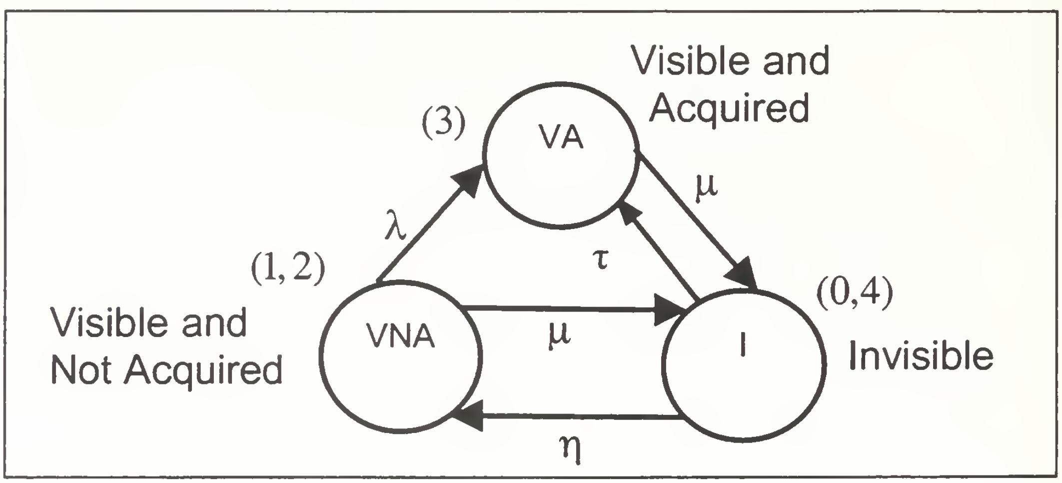

Figure 11. The levels of target discrimination output in the modified JANUS detection file corresponding to the states in the Markov chain model of target acquisition if the identification firing criterion is used. Targets discriminated at a level of discrimination of three are acquired and will be fired upon. Thus, in identification, fewer number of targets are acquired to fire at.

The Markov chain model of target acquisition in Figure 9 has the following Forward-Kolmogorov Equations:

$$\begin{cases} \dfrac{dp_I}{dt} = -(\tau+\eta)\, p_I + \mu\, p_{VNA} + \mu\, p_{VA} \\ \dfrac{dp_{VNA}}{dt} = \eta\, p_I - (\lambda+\mu)\, p_{VNA} \\ \dfrac{dp_{VA}}{dt} = \tau\, p_I + \lambda\, p_{VNA} - \mu\, p_{VA} \end{cases} \tag{4.25}$$

where,

p_I = probability of a target becoming invisible

p_{VNA} = probability of a target becoming visible and not acquired

p_{VA} = probability of a target becoming visible and acquired

When a target becomes VA, it will be fired upon by the JANUS direct-fire algorithms. Setting the left-hand sides which represent rates of change in equations (4.25) equal to zero to obtain:

$$\begin{cases} (\tau+\eta)p_I = \mu\, p_{VNA} + \mu\, p_{VA} \\ (\lambda+\mu)p_{VNA} = \eta\, p_I \\ \mu\, p_{VA} = \tau p_I + \lambda\, p_{VNA} \end{cases}$$

$$p_I + p_{VNA} + p_{VA} = 1 \tag{4.26}$$

Successive substitution of the system of equations in (4.26) into each other yields the following steady state equation:

$$p_{VA}(\infty) = \frac{\eta\lambda + \tau(\lambda+\mu)}{(\tau+\eta+\mu)(\lambda+\mu)} \tag{4.27}$$

Each rate is calculated as follows. For example, $\lambda=(1/[(1/n)^*\Sigma\ (t_{VA} - t_{VNA})])$. That is for a T72 observing an M1A1, λ is the transition rate from VNA to VA of the M1A1 tank. The value n is the total number of transitions for that particular M1A1 from VNA to VA. t_{VNA} is the time of entering the VNA state for that particular M1A1 tank. t_{VA} is the time of entering the VA state for that particular M1A1 tank. The target availability values, A_{ij} and B_{ji}, in heterogeneous force equations are equal to the value of $p_{VA}(\infty)$ for given i and j pairs of observers and targets. After the values of A_{ij} and B_{ji} are calculated using the steady state solution of the continuous time three state Markov chain model of target acquisition, the conditional kill rates, α_{ij} and β_{ji}, can easily be calculated using the maximum likelihood estimators given by (4.21) and (4.22). The target priority values required in those estimator equations are manually extracted from JANUS combat systems database files for every firer-target combination. They are provided in Table 10 and Table 11 in Chapter V.

2. Variance of the Maximum Likelihood Estimates

The variance of the COMAN MLE is affected by the density of the time series data generated as a result of the Monte Carlo simulation and the spread of the data points. If too few data points are available for the COMAN MLE approach, because there were insufficient casualties of a certain entity, the variance of the MLE will exceed a relevant range [Ref. 7]. A denser time series data could reduce the variance of the MLEs since the variance between the times of kills is zero if the times of occurrence of casualties are equally spread out. However, in a given battle, casualties occur in groups due to the

different ranges of enemy weapon systems and the maneuver plan selected. Thus, in practice the times of casualties cannot be equally spread out to reduce the variance of the MLEs.

One approach used in a previous thesis [Ref. 17] to overcome this time series data problem was to perform multiple runs of the same scenario in JANUS with different random number seeds. The time series data obtained from multiple independent runs was combined into a single file to obtain denser time series data [Ref. 17]. This approach is reasonable since combining results of ten independent replications of the same scenario into a single file is analogous to summing ten independent and identically distributed (i.i.d.) random variables, which would result in a new random variable. However, this approach is inappropriate because it would result in unrealistic, inflated conditional kill rate estimates for the engagement type being simulated.

This thesis uses a different approach to reduce the variance for the MLE while providing realistic conditional kill rate estimates. Ten replications of the same scenario are run with different random number seeds. MLEs of conditional kill rates for all weapon system combinations are computed for each replication. These ten estimates of the conditional kill rates are then added and divided by ten. The resulting averaged MLE of the attrition rates will have one tenth of the variance of the MLE of a single replication. To explain this fact more clearly, assume that $x_1,\ldots,x_{10}$ are ten i.i.d. realizations of the random variable X. $\bar{x} = (\sum X_i) / 10$ is the average of ten MLEs. Then, the variance of $\bar{x}$ is $\mathrm{Var}(\bar{x}) = \mathrm{Var}(\sum X_i / 10) = \sum \mathrm{Var}(X_i) / 100$. Since each replication is independent,

the covariance terms are zero. Thus, the resulting variance of the averaged MLE is one tenth of that of a single replication. In addition, the attrition rate estimate will still be a realistic one.

The independence of each replication is a necessary condition for the procedure explained above. If a different random number seed is used for each replication and there is no man-in-the-loop, the randomness of each replication occurs as a result of the Monte Carlo processes on random variables, i.e., kill and detection events. Thus, each replication is independent and will have different results.

The approach mentioned above for the estimation of conditional kill rates is also used for the estimation of target availability parameters. That is, target availability parameters obtained from the previously mentioned JAVA program are added and their average is taken.

3. Confidence Intervals for the Maximum Likelihood Estimates

Rice [Ref. 26] discusses three methods for obtaining confidence intervals for MLEs: exact methods, approximations based on the large sample properties, and bootstrap confidence intervals. The exact methods are usually the exception in practice since detailed knowledge of the sampling distribution is usually not available. Approximations based on large sample properties of MLEs are not always reasonable because of small sample sizes. Thus, a bootstrap confidence interval for the estimates of target availability and attrition-rate estimates is the best choice for use in this research. [Ref. 26]

Bootstrap is best explained by Bradley Efron in his book "An Introduction to the Bootstrap" [Ref. 27]. Bootstrap is based on the notion of a bootstrap sample. One starts with n data points assumed to be modeled by i.i.d. draws from a distribution function F. The emprical distribution using the data is constructed. A bootstrap sample is a random sample of size n drawn from the empirical distribution $\hat{F}$. Suppose $\mathbf{x}^* = (x^*_1, x^*_2, \ldots, x^*_n)$ is a random sample of size n drawn from $\hat{F}$. The star notation indicates that $\mathbf{x}^*$ is not the actual data set $\mathbf{x}$, but a resampled version of $\mathbf{x}$. That is, the sample $\mathbf{x}^*$ is a random sample of size n drawn with replacement from the population of n objects $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. We might have $x^*_1 = x_5$, $x^*_2 = x_7$, ..., $x^*_n = x_5$. Thus, the bootstrap data set $\mathbf{x}^*$ consists of the members of the original data set $\mathbf{x}$.

We wish to find the confidence interval for a parameter of interest θ. [Ref. 27] B bootstrap resamplings are created as explained above. Then, the $(1\text{-}2\alpha)$ percentile interval for the bootstrap can be written as:

$$\left[\hat{\theta}_{\%,lo}, \hat{\theta}_{\%,up}\right] \approx \left[\hat{\theta}^{*(\alpha)}_{B}, \hat{\theta}^{*(1-\alpha)}_{B}\right] \tag{4.28}$$

where, B is some finite number of resamplings. The standard normal and percentile intervals should agree if the bootstrap distribution of $\hat{\theta}^*$ is nearly normal. According to the central limit theorem, as the size of the original data set approaches infinity, the bootstrap sampling distribution of the parameter estimated should look Gaussian. However, Efron says it may look very non-normal for small original data sets. [Ref. 27]

Bootstrap confidence intervals are developed in Chapter V for the mean of attrition-rate estimates. For example, ten estimates obtained as a result of the MLE (4.21) for a T-72 firing on an M1A1 tank form the original data set explained above. We are interested in the mean as our statistic since we come up with point estimates by taking the mean of the results of 10 runs for each weapon combination. 1000 bootstrap resamples of ten are created from this sample with replacement and their means are calculated using the bootstrap commands found in S-PLUS. The 975^{th} and the 25^{th} ranked values of the means of 1000 resamples of ten form a 95% confidence interval around the calculated average. In addition, even though the target availability values calculated, by finding the steady-state solution of the Markov chain model, do not result from a maximum likelihood estimator, a bootstrap confidence interval can still be used to determine the confidence interval around the mean of each of ten computed values of target availability for A and B as explained above.

This chapter has shown the development and use of a MLE approach for the cases of homogeneous and heterogeneous forces. Chapter V will provide the application of the MLE approach to the time series data files generated as a result of JANUS runs. The Lanchester attrition-rate Phase I estimates obtained from the MLEs will then be used in the aggregate ATCAL Phase II attrition equations in Chapter V. The results of the aggregate replay equations will then be compared to the attrition results of JANUS runs for different firing criteria to see if the equations are calibrated.

V. ANALYSIS AND RESULTS

A. DETERMINING TARGET AVAILABILITY PARAMETERS

Each of the twenty detection files obtained from the JANUS simulation runs were used by the JAVA program in Appendix C to calculate the steady state values of the three-state Markov chain model of target acquisition. The JAVA program in Appendix C imports a graph and networks package called König [Ref. 28] to assist in the data reduction effort. Entities in the simulation are treated as the nodes of a network in König. Interaction between entities, such as the detection and kill events as used in this study, represent the arcs between the nodes. Each arc can have several attributes that represent the entity's state. Examples of these attributes are the time of kill, the time of detection, and the level of discrimination of an entity in the simulation. Another JAVA package called Swing [Ref. 29] was also imported to create the frames to display the information obtained from JANUS output files.

Each of the twenty detection files, such as the one provided in Table 2, obtained from twenty runs of JANUS Scenario 150 were in different lengths. The smallest file was 1366 lines long for run 15030 in a JANUS simulation run time of 16:45 minutes. The largest file was 2796 lines long for run 15048 in a JANUS simulation run time of 18:20 minutes. Rows of information were organized in four columns for each file. These modified detection files were then used by the JAVA program in Appendix C to calculate the rates and the steady state values of the Markov chain model explained in Chapter IV.

This program calculates the target availability for each of the 170 entities used in the simulation. Then, it provides a steady state value, which corresponds to the target availability as explained in Chapter IV, for each weapon system type. For example, the program calculates the target availability values for each of the 56 M2IFVs as observed by each of the eight T-72 tanks. Then, these values are combined to calculate a single target availability value of all M2IFV targets for the T-72 tanks. This value is displayed in a frame such as in Figure 13 and then manually input into an EXCEL spreadsheet for use in future calculations of the conditional kill rate estimates. Processing runs of the detection files through the JAVA program in Appendix C took between 35 to 45 minutes on a Pentium II 266 MHz. personal computer. Two sample outputs of the program in Appendix C are provided in Figure 12 and Figure 13. The resulting target availability values of each run for all direct-fire weapons' firer-target combinations are provided in Appendix D.

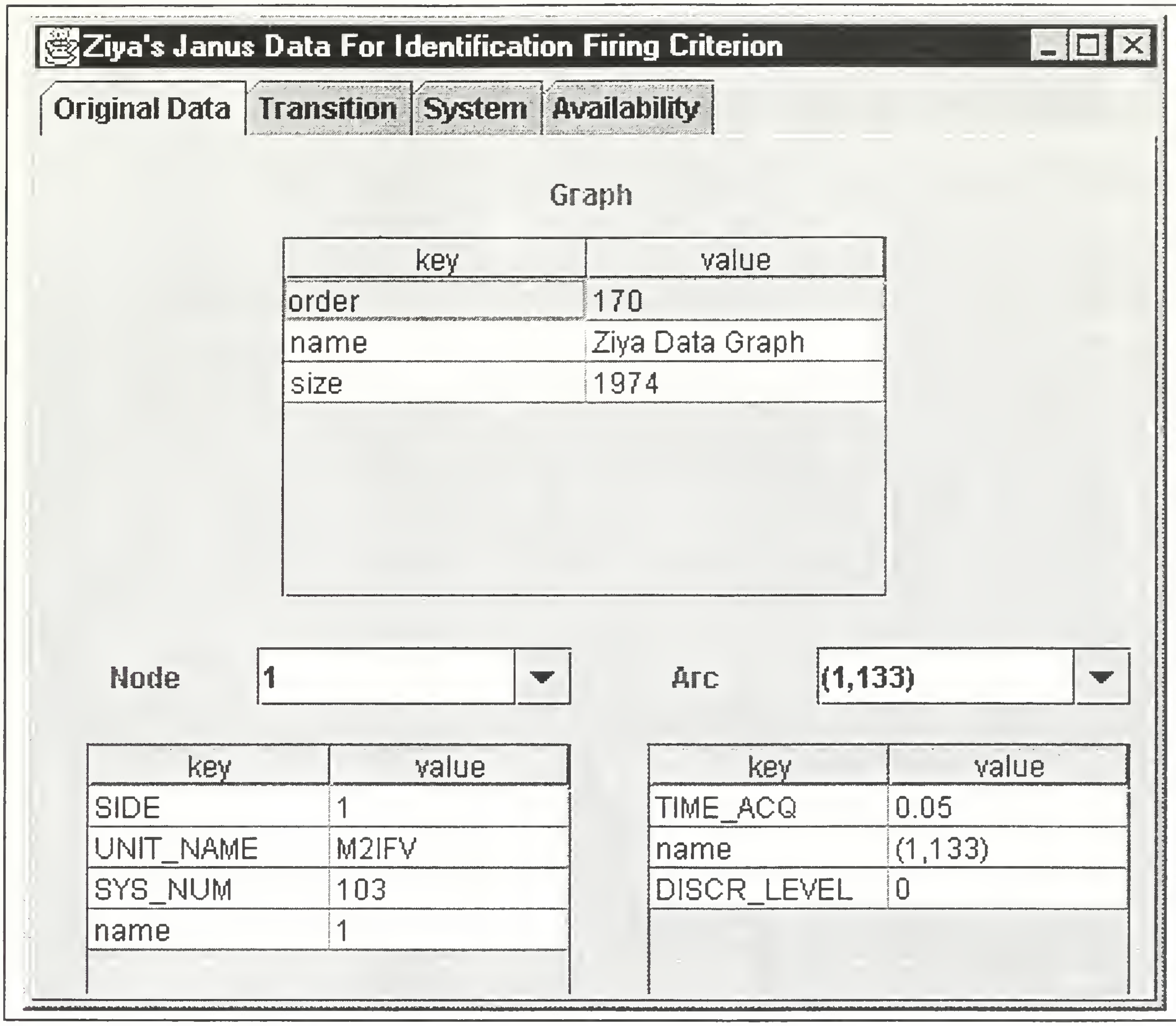

Figure 12. A sample output of the JAVA program used in data reduction. This figure shows the detection file information read in by the program. There are 170 nodes corresponding to the number of weapon systems present in the simulation. There are 1974 arcs, corresponding to the number of detection events that took place between opposing forces when run 15049 of the JANUS scenario was made. The information on the arcs are the time of acquisition and the discrimination level of the detection event. The weapon system that made the detection is the 1st element of the M2IFVs. The user can use the drop-down lists to display information concerning other weapon systems.

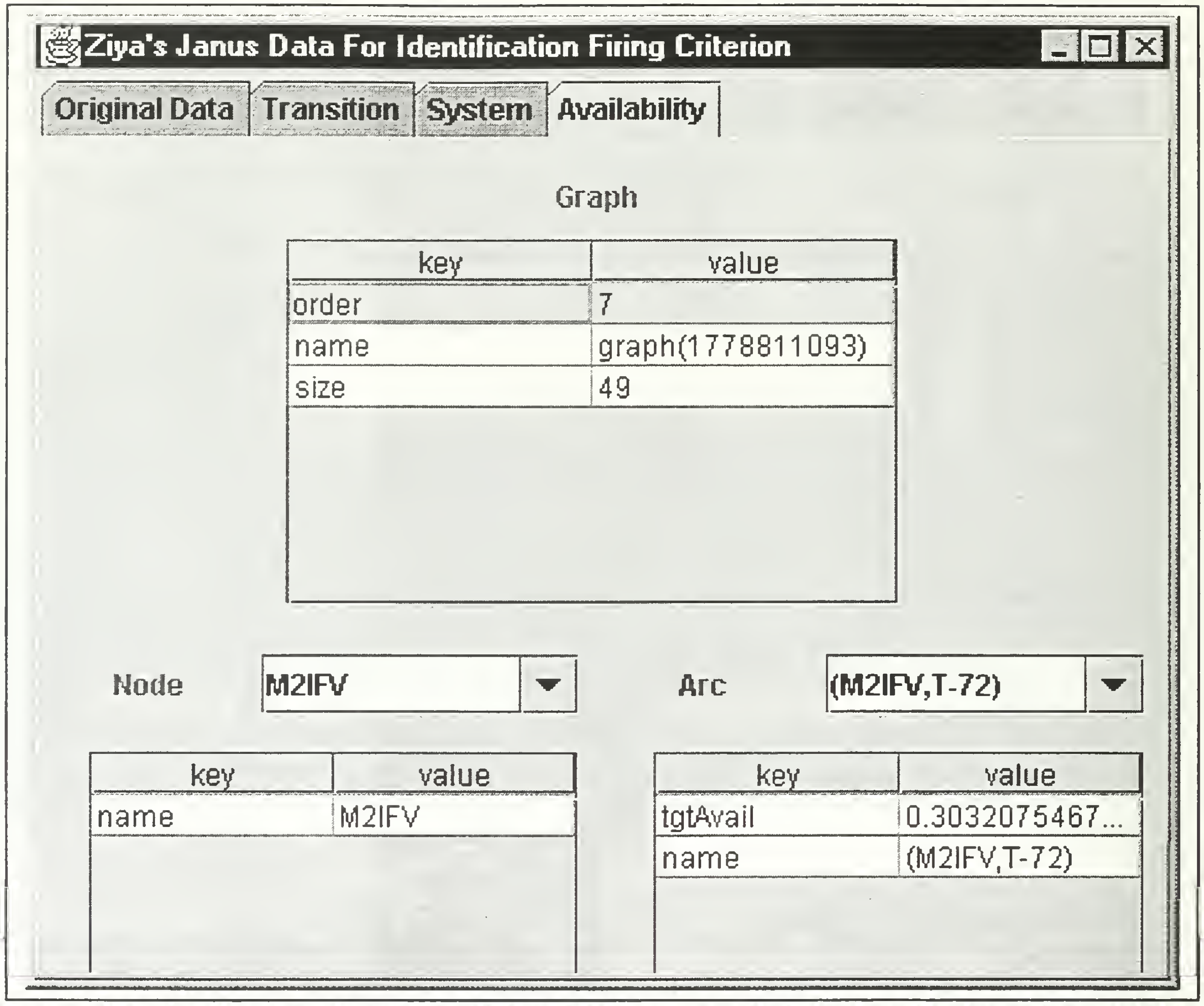

Figure 13. Another sample output of the JAVA program that displays final target availability information for all M2IFVs of all T-72 tanks. The menu item "Transition" on top of the figure displays the menu where the transition rates were calculated for each entity. The menu item "System" displays the information on total systems transition rates for weapon types such as M1A1 tanks. The menu item Original Data is displayed in Figure 12.

The final target availability values for the average of the 10 runs for each weapon system combination for a given firing criterion, are provided in Tables 5, 6, 7 and 8. Tables 5 and 6 contain the averaged target availability values for the recognition firing criterion. Tables 7 and 8 contain the target availability values for the identification firing

criterion. These target availability values are used to determine the attrition-rate parameters.

Detected Units	Observing Units			
	T-72	T-80	BMP	BMP-2
M1A1	0.2594	0.2372	0.1800	0.0647
M2IFV	0.5834	0.2488	0.0998	0.1831

Table 5. Final A_{ij} values for the recognition firing criterion.

Detected Units	Observing Units	
	M1A1	M2IFV
T-72	0.4861	0.3771
T-80	0.2358	0.2966
BMP	0.19	0.2228
BMP-2	0.2709	0.1816

Table 6. Final B_{ji} values for the recognition firing criterion.

Detected Units	Observing Units			
	T-72	T-80	BMP	BMP-2
M1A1	0.3457	0.1976	0.3022	0.142
M2IFV	0.5035	0.4096	0.2363	0.3334

Table 7. Final A_{ij} values for the identification firing criterion.

Detected Units	Observing Units	
	M1A1	M2IFV
T-72	0.6294	0.3063
T-80	0.3347	0.2091
BMP	0.1848	0.3366
BMP-2	0.4877	0.3011

Table 8. Final B_{ji} values for the identification firing criterion.

B. DETERMINING ATTRITION-RATE PARAMETERS

Determination of the attrition-rate parameters requires the substitution of target availability data values from Tables 5, 6, 7, and 8 into equations (4.21) and (4.22) for α_{ij}

and β_{ji} respectively. Twenty direct-fire kill events' files obtained from JANUS output were used to obtain the times between kills and the numbers of targets and firers available at any time of kill during the simulation. An extract from a modified direct-fire kill file is provided in Table 9. Each file contains 57 lines that represent the combined total of entities killed from both sides in the simulation considered for analysis as explained in Chapter IV.

Firer Unit ID Number	Target Unit ID Number	Time Of Kill
10	143	0.7398
10	136	1.1994
140	5	1.2305
140	6	3.2203
133	45	5.1222
150	43	8.5133
142	46	8.8689
151	44	11.0057
138	20	11.1208
146	109	11.5665
134	107	11.6275
138	19	11.8325
92	146	11.8767
114	137	12.1798
138	23	12.2122
156	41	12.2253
139	108	12.3062

Table 9. A part of a sample kill file obtained from run 15049 of the JANUS scenario.

The JAVA program in Appendix E was used to read the direct-fire kill files and to keep track of the times between kill events and the numbers of remaining entities in the simulation.

This program also displays the information contained in direct-fire kill files in a frame such as the one provided in Figure 12. The results were written to a text file, which were then imported into a commercial spreadsheet, EXCEL, for further analysis. Processing of each of the direct-fire kill files using the JAVA program in Appendix E took between one to two minutes on a Pentium II 266 MHz. personal computer. The following priority lists were substituted into equations (4.21) and (4.22) in the EXCEL spreadsheet along with the target availability values in Tables 5, 6, 7, and 8, and the other data mentioned above, i.e., the number of remaining entities, as appropriate. The priority lists in Table 10 and Table 11 are used for both criteria.

Target Units	Firing Units			
	T-72	T-80	BMP	BMP-2
M1A1	2	2	1	1
M2IFV	1	1	2	2

Table 10. Priority of X (Blue) Targets for Y (Red) Firers

Target Units	Firing Units	
	M1A1	M2IFV
T-72	3	4
T-80	4	3
BMP	1	1
BMP-2	2	2

Table 11. Priority of Y (Red) Targets for X (Blue) firers

The conditional kill rate estimates calculated for each run in the EXCEL spreadsheet are in Appendix F. The final attrition rate estimates are a result of the average of 10 estimates for each run are in Tables 12 and 13 for recognition firing criterion and in Tables 14 and 15 for identification firing criterion. The α_{ij} and β_{ji} values

for recognition firing criterion are displayed in Tables 12 and 13 respectively. The α_{ij} and β_{ji} values for identification firing criterion are displayed in Tables 14 and 15 respectively.

Target Units	Firing Units			
	T-72	T-80	BMP	BMP-2
M1A1	0.1202	0.0501	0.0356	0.0462
M2IFV	0.0582	0.0194	0.0620	0.0694

Table 12. The attrition-rate coefficients α_{ij} values for recognition firing criterion.

Target Units	Firing Units	
	M1A1	M2IFV
T-72	0.0877	0.0003
T-80	0.0037	0.0008
BMP	0.0000	0.0000
BMP-2	0.3236	0.1470

Table 13. The attrition rate coefficients β_{ji} values for recognition firing criterion.

Target Units	Firing Units			
	T-72	T-80	BMP	BMP-2
M1A1	0.1111	0.0477	0.0268	0.0324
M2IFV	0.0508	0.0264	0.0529	0.0353

Table 14. The attrition-rate coefficients α_{ij} values for identification firing criterion.

Target Units	Firing Units	
	M1A1	M2IFV
T-72	0.1556	0.0005
T-80	0.0019	0.0017
BMP	0.0000	0.0000
BMP-2	0.4104	0.0659

Table 15. The attrition rate coefficients β_{ji} values for identification firing criterion.

BMPs do not get killed by BLUE in any of the runs of the scenario in either firing criteria. However, BMPs have a positive attrition rate and do kill Blue weapon systems. The reason for this is the location of the BMPs in the scenario terrain. They are in the

rear compared to other Red direct-fire weapon systems and on defensible terrain with very clear lines of fire against the Blue side.

C. CONFIDENCE INTERVALS FOR TARGET AVAILABILITY AND ATTRITION-RATE PARAMETERS

The following 95 % bootstrap confidence intervals have been obtained for the target availability and the attrition-rate coefficients for both firing criteria.

1. Confidence Intervals for Target Availability Parameters

We are 95% certain that the target availability values calculated by the JAVA program in Appendix C are contained in the intervals given below by the bootstrap confidence intervals. The confidence intervals for the recognition firing criterion are listed in Tables 16 and 17. The confidence intervals for the identification firing criterion are listed in Tables 18 and 19.

Detected Units	Observing Units			
	T-72	T-80	BMP	BMP-2
M1A1	[0.1567 - 0.3932]	[0.1181- 0.4194]	[0.1217 -0.2476]	[0.03439 -0.1016]
M2IFV	[0.4148 -0.7873]	[0.08037-0.4685]	[0.05353 –0.147]	[0.03636 -0.4536]

Table 16. Bootstrap confidence intervals of A_{ij} values for the recognition firing criterion.

Detected Units	Observing Units	
	M1A1	M2IFV
T-72	[0.3325- 0.6372]	[0.224 - 0.5344]
T-80	[0.09136 - 0.4518]	[0.09106 - 0.5596]
BMP	[0.07765 - 0.376]	[0.1481 - 0.3366]
BMP-2	[0.1808 - 0.3543]	[0.1078 - 0.2523]

Table 17. Bootstrap confidence intervals of B_{ji} values for the recognition firing criterion.

Detected Units	Observing Units			
	T-72	T-80	BMP	BMP-2
M1A1	[0.2547 -0.4292]	[0.09216 -0.3255]	[0.1998 - 0.3936]	[0.08977 -0.1964]
M2IFV	[0.3365 -0.6926]	[0.2153 -0.5897]	[0.1696 - 0.2964]	[0.1391 -0.5441]

Table 18. Bootstrap confidence intervals of A_{ij} values for the identification firing criterion.

Detected Units	Observing Units	
	M1A1	M2IFV
T-72	[0.5807 - 0.6961]	[0.2127 - 0.4106]
T-80	[0.1897 - 0.5208]	[0.07364 - 0.3748]
BMP	[0.1048 - 0.31]	[0.1699 - 0.5098]
BMP-2	[0.3883 - 0.5993]	[0.1667 - 0.4653]

Table 19. Bootstrap confidence intervals of B_{ji} values for the identification firing criterion.

2. Confidence Intervals for Attrition-Rate Parameters

We are 95% certain that the target availability values calculated by the JAVA program in Appendix E are contained in the intervals given below by the bootstrap confidence intervals. The confidence intervals for the recognition firing criterion are listed in Tables 20 and 21. The confidence intervals for the identification firing criterion are listed in Tables 22 and 23.

Target Units	Firing Units			
	T-72	T-80	BMP	BMP-2
M1A1	[0.1004 - 0.1397]	[0.04148 -0.05729]	[0.03092 -0.04055]	[0.03215 -0.05834]
M2IFV	[0.04179 - 0.07427]	[0.01507 -0.02407]	[0.05221 - 0.07157]	[0.05368 - 0.08781]

Table 20. Bootstrap confidence intervals of the attrition-rate coefficients α_{ij} for recognition firing criterion.

Target Units	Firing Units	
	M1A1	M2IFV
T-72	[0.07829 -0.09782]	[0 - 0.00081]
T-80	[0.00333 - 0.00404]	[0 - 0.00181]
BMP	0	0
BMP-2	[0.2679 - 0.3766]	[0.08713 - 0.2034]

Table 21. Bootstrap confidence intervals of the attrition rate coefficients β_{ji} for recognition firing criterion.

Target Units	Firing Units			
	T-72	T-80	BMP	BMP-2
M1A1	[0.09008 - 0.1313]	[0.03685 - 0.06165]	[0.02108 -0.03244]	[0.02305 -0.04204]
M2IFV	[0.04149 - 0.05945]	[0.02111 -0.03214]	[0.04369 - 0.0616]	[0.02535-0.04557]

Table 22. Bootstrap confidence intervals of the attrition-rate coefficients α_{ij} for identification firing criterion.

Target Units	Firing Units	
	M1A1	M2IFV
T-72	[0.1323 -0.1776]	[0.00014 - 0.00097]
T-80	[0.00117 -0.00299]	[0.00034 - 0.003762]
BMP	0	0
BMP-2	0.2834 - 0.5689	0.03443 - 0.09731

Table 23. Bootstrap confidence intervals of the attrition rate coefficients β_{ji} for identification firing criterion.

D. REPLAY MODEL FOR ATCAL PHASE II

The replay of the attrition results for the same size force-mix used in JANUS simulation runs is achieved here by using the set of equations (2.14) of the heterogeneous force ATCAL Phase II model. All of the model parameters have been estimated and provided in Tables 5-8, in Tables 10 and 11, and in Tables 12-15. These parameters are substituted into the set of equations (2.14) and the values of these equations are computed using an EXCEL spreadsheet. The attrition results of this aggregate model are compared

to the attrition results of JANUS run 15040 for the identification firing criterion. Similarly, attrition results using the ATCAL set of equations (2.14) are compared to the attrition results of JANUS run 15028 for the recognition firing criterion. Figures 14 and 15 depict the decrease in force sizes as a result of the high-resolution battle versus the decrease as a result of the aggregate replay model. For the identification firing criterion, Figure 14 is provided. For the recognition firing criterion, Figure 15 is provided.

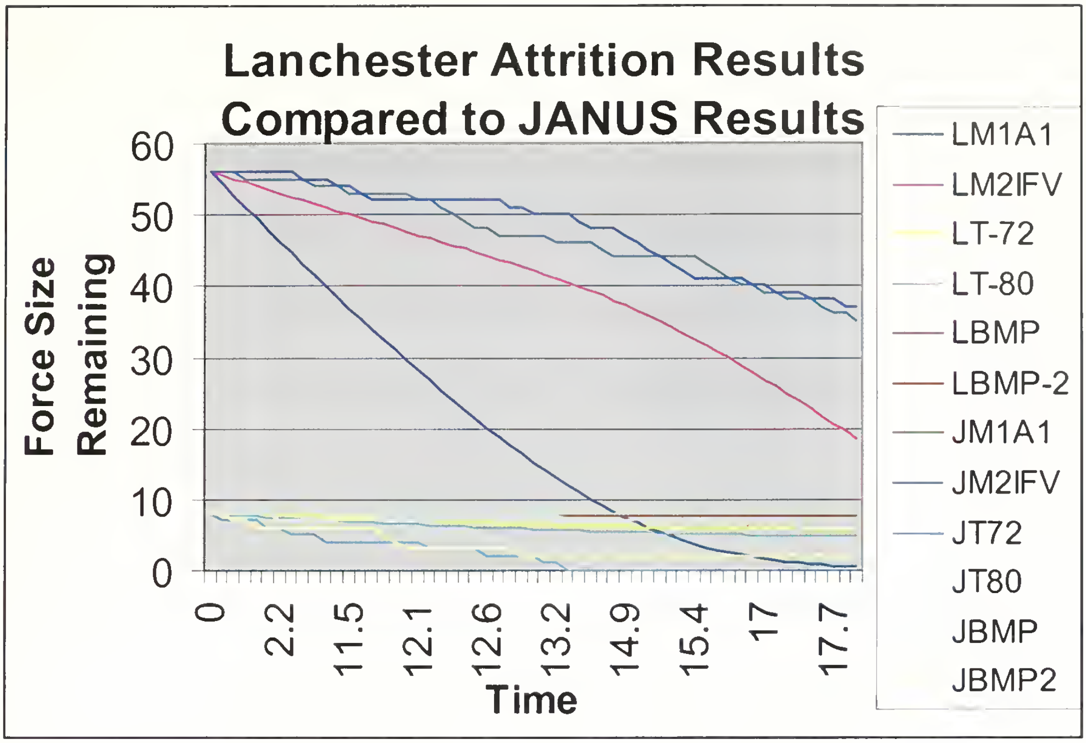

Figure 14. The graph displays the decrease in force size obtained from the ATCAL replay equations (2.14) compared to the high-resolution simulation results obtained from JANUS run 15040 using identification firing criterion. Weapon names in the legend starting with L, such as LM1A1, indicate ATCAL replay results. Weapon names starting with J, such as JM1A1 indicate JANUS results.

The decrease in force size of Red weapon systems as a result of the ATCAL attrition model and also as a result of attrition in JANUS simulation run 15040 using the identification firing criterion are shown by the curves in the bottom part of Figure 14. The results of the aggregate model closely resemble JANUS attrition results for all Red weapon systems.

The initial number of weapons for the Blue side was 56 for M1A1s and the M2IFVs. The decrease in force size for these Blue weapon systems in JANUS and also as a result of the ATCAL attrition model are shown by the four curves in the upper part of

the graph. The results of the aggregate model for the M1A1 weapon system seem to closely agree with the attrition results from JANUS run 15040. However, the same is not true for the M2IFV weapon system. The attrition results are not as close as they were for the M1A1 system. Time constraints on this study have not permitted a complete investigation of this issue with the M2IFV weapon system. However, comparison of aggregate results to other JANUS scenario runs might return results closer to the results of the aggregate model.

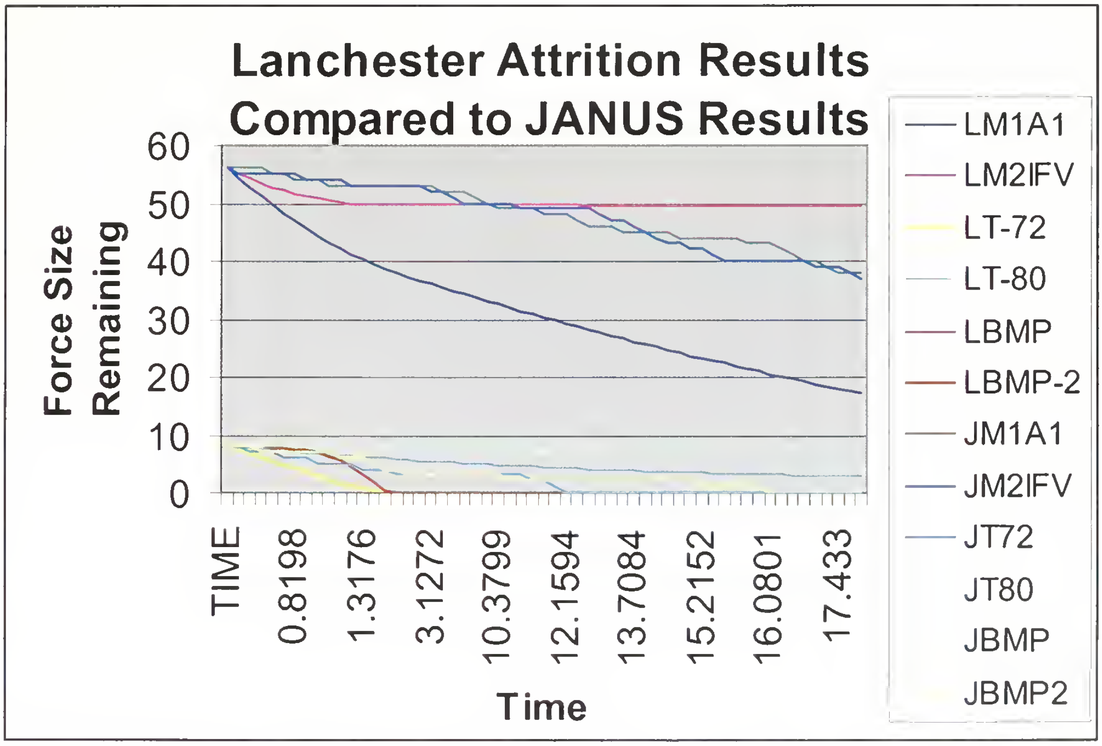

Figure 15. The graph displays the decrease in force size obtained from the ATCAL replay equations (2.14) compared to the high-resolution simulation results obtained from JANUS run 15028 using recognition firing criterion. Weapon names in the legend starting with L, such as LM1A1, indicate ATCAL replay results. Weapon names starting with J, such as JM1A1 indicate JANUS results.

The attrition results of the aggregate ATCAL Phase II model of attrition and the attrition results of the JANUS simulation run 15028 of the recognition firing criterion

appear in Figure 15. Once again, the results of the aggregate model are not as close to JANUS attrition results for the M2IFV weapon system as other systems. Time constraints on this study have not permitted a complete investigation of this issue with the M2IFV weapon system. However, comparison of aggregate results to other JANUS scenario runs might return results closer to the results of the aggregate model.

As a result, except for the M2IFV weapon system, all of the attrition equations obtained by using the COMAN MLE approach are calibrated for use in aggregate models that use ATCAL attrition equations. Recognition firing criterion results are similar to the identification firing criterion results.

appear in Figure 15. Once again, the results of the aggregate model are not as close to JANUS attrition results for the M2IFV weapon system as other systems. Time constraints on this study have not permitted a complete investigation of this issue with the M2IFV weapon system. However, comparison of aggregate results to other JANUS scenario runs might return results closer to the results of the aggregate model.

As a result, except for the M2IFV weapon system, all of the attrition equations obtained by using the COMAN MLE approach are candidates for use in aggregate models that use ATCAL attrition equations. Recognizing firing criterion issues are central to the identification firing criterion issues.

VI. CONCLUSIONS AND RECOMMENDATIONS

This thesis demonstrates the application of a sound mathematical formulation and a maximum likelihood estimation approach to determine Lanchester attrition-rates for direct fire engagements of heterogeneous force mixes. This is proposed as an alternative method of analysis that could potentially be used by CAA in calculation of model parameters during ATCAL Phase I. This study is a first in its use of a three-state, continuous-time Markov chain model of target acquisition to determine target availability parameters of an aggregate attrition model. This research also explains the theoretical basis of ATCAL and its origins as it relates to Gordon Clark's research in 1969.

An issue that is beyond the scope of this research is the application of a COMAN MLE approach to estimation of ATCAL model parameters for indirect fire engagements such as mortars and artillery. Specifically, it would be interesting to see how proximity effects of indirect fire weapons and the resulting collateral damage would be modeled by using ATCAL equations and JANUS. This was not possible to do with JANUS Version 6.88 since it does not provide detailed collateral damage information. Even if some of the routines were modified, as in the case of the modified detection file obtained in this thesis, JANUS does not internally compute any collateral damage information in Version 6.88. However, this may be possible to do with the newest JANUS Version 7.0 that provides more detailed information on probabilities of kill for crews and passengers of

vehicles due to effects of weapons fired at vehicles. [Ref. 30] Such an undertaking is recommended for future research.

Another feature of this research that might be improved is the complete automation of the computer programs. The JAVA programs reduce the simulation data, compute the COMAN / ATCAL model parameters and output these results into a file or into EXCEL for further analysis. Even though the programs handle a very large amount of simulation data and reduce them to manageable and meaningful pieces, they are not fully automated and some manual user labor is required to input some of the reduced simulation data into the EXCEL spreadsheets for analysis. Possible future work in this area might address this issue of complete automation of all the data input from JANUS and the automatic calculation of model parameters without any manual effort.

In order to obtain useful attrition-rate coefficients and target availability information, the user would have to run many scenarios in different terrains, using different types of engagements, such as defense, or retrograde operations, etc., and different force mixes. This would help form a library of attrition-rate coefficients and target availability values that could immediately be used in an aggregate model to analyze results of operations plans when future contingencies arise.

LIST OF REFERENCES

1. Taylor, James G. *Force-On-Force Attrition Modeling.* Military Applications Section of the Operations Research Society of America., Arlington, Virginia, 1980.

2. Taylor, James G., Buss, A.H. et al. *Redesign of ATCAL Attrition Algorithm for Military OR Problems in the Twenty First Century.* 66th MORS Symposium. Naval Postgraduate School. Monterey, CA. June 25, 1998.

3. Analysis Support Directorate. *ATCAL: An Attrition Model Using Calibrated Parameters.* Technical Paper No: CAA-TP-83-3.U.S. Army Concepts Analysis Agency. Bethesda, Maryland. August, 1983.

4. U.S. Army Concepts Analysis Agency (CAA), *Concepts Evaluation Model VI (CEM VI), Volume I-Technical Description*, Bethesda, MD, January 1985 (revised October 1987).

5. Appleget, Jeffrey A. *The Combat Simulation of Desert Storm with Applications for Contingency Operations*, in Jerome Bracken, Moshe Kress, Richard E. Rosenthal, eds., *Warfare Modeling.* Military Operations Research Society (John Wiley and Sons, Inc.) 1995.

6. Science Applications International Corporation (SAIC), *Integrated Theater Engagement Model (ITEM) Technical Manual Version 8.0.* San Diego, CA August 15, 1998.

7. Gordon M. Clark. *The Combat Analysis Model.* Department of Industrial Engineering. Columbus, Ohio, 1969.

8. Taylor, James G. *Underlying Lanchester-Type Model for ATCAL, Working Paper #3*, SAC of OSD PA & E Project. Monterey, CA. July 23, 1997.

9. Simulation, Training and Instrumentation Command. *JANUS Software User's Manual Version 6.3 UNIX Model.* Orlando, Florida (undated).

10. Mansager, Bard and Taylor, James G. *Investigation of the Attrition-Calibration (ATCAL) Approach for Modeling Joint Warfare.* Proposal for Institute of Joint Warfare Analysis Research. Monterey, CA. October, 1997.

11. Vandiver, E.B. *Letter from E.B. Vandiver (Director, U.S. Army Concepts Analysis Agency) to A.W. Marshall (Director of Net Assessment, Office of the Secretary of Defense).* 9 April 1992.

12. Taylor, James G. *A Tutorial on the Determination of Single-Weapon-System-Type Kill Rates for Use in Lanchester-Type Combat Models.* August, 1982.

13. Taylor, James G. *Lanchester Models of Warfare, Volume I.* March, 1983.

14. Fowler, Bruce W. *An Introduction to Lanchestrial Attrition Mechanics.* ITT Research Institute, Chicago, IL. September, 1996.

15. Taylor, James G. *Development of Basic ATCAL Assessment Equations from Underlying Lanchester-Type Model*, *Working Paper #4*, SAC of OSD PA & E Project. Monterey, CA. July 25, 1997.

16. Taylor, James G. *The Exponential Function and Its Average*, *Working Paper #1*, SAC of OSD PA&E Project, Operations Research Department, Naval Postgraduate School, Monterey, CA, July 1997.

17. Park, Hun Keun. *Analysis of Air Land Combat Tactics Using JANUS (T) System.* Master's Thesis, Naval Postgraduate School, Monterey, CA. March 1990.

18. AMSAA Combat Integration Division. *The Compendium of Close Combat Tactical Trainer Algorithms, Data, Data Structures and Generic System Mappings.* U.S. Army Material Systems Analysis Activity. Aberdeen Proving Ground, Maryland. June 1996.

19. Parish, Randall M. and Kellner, Alvin D. *Target Acquisition Algorithm in JANUS Army.* TRADOC Analysis Command, White Sands Missile Range, NM, October 1992.

20. Parish, Randall M. E-mail to 1LT. Yildirim. *SUBJECT: Target Acquisition in JANUS Army.* From TRADOC Analysis Command, White Sands Missile Range, NM. 21 January 1999.

21. Kellner, Alvin D. *MEMORANDUM FOR RECORD, SUBJECT: NVEOD Detection in JANUS (A).* TRADOC Analysis Command, White Sands Missile Range, NM. 14 July, 1992.

22. Larimer, Larry R., *Building an Object Model of a Legacy Simulation*. Master's Thesis, Naval Postgraduate School, Monterey, CA. June 1997.

23. Larsen, Richard J. and Marx, Morris L. *An Introduction to Mathematical Statistics and its Applications.* Prentice-Hall, Englewood Cliffs, New Jersey. 1986.

24. Taylor, James G. *Maximum Likelihood Estimation of COMAN / ATCAL Parameters, Working Paper #1*. JTASC of ACOM Project. Monterey, CA. March 21, 1998.

25. Taylor, James G. *Lanchester Models of Warfare, Volume II*. March, 1983.

26. Rice, John A. *Mathematical Statistics and Data Analysis*. 2nd ed. Wadsworth Publishing Company, 1995.

27. Efron, Bradley and Tibshirani, Robert J. *An Introduction to the Bootstrap*. Chapman & Hall, New York, NY. 1993.

28. Jackson, LeRoy "Jack". *König Beta 1.0 User Guide.* [http://www.trac.nps.navy.mil/jackson1/könig]. 1998.

29. Sun Microsystems Inc., Homepage. *Swing.* [http://www.java.sun.com/products/jfc/tsc/what_is_swing]

30. Simulation, Training and Instrumentation Command (STRICOM). *Release Notes For JANUS UNIX Version 7.0*. Orlando, FL. 4 September 1997.

APPENDIX A. JANUS SCREEN I

The analyst is allowed to specify a target list update cycle time in JANUS Screen I that is longer than the detection cycle time. This influences performance in a given JANUS scenario.

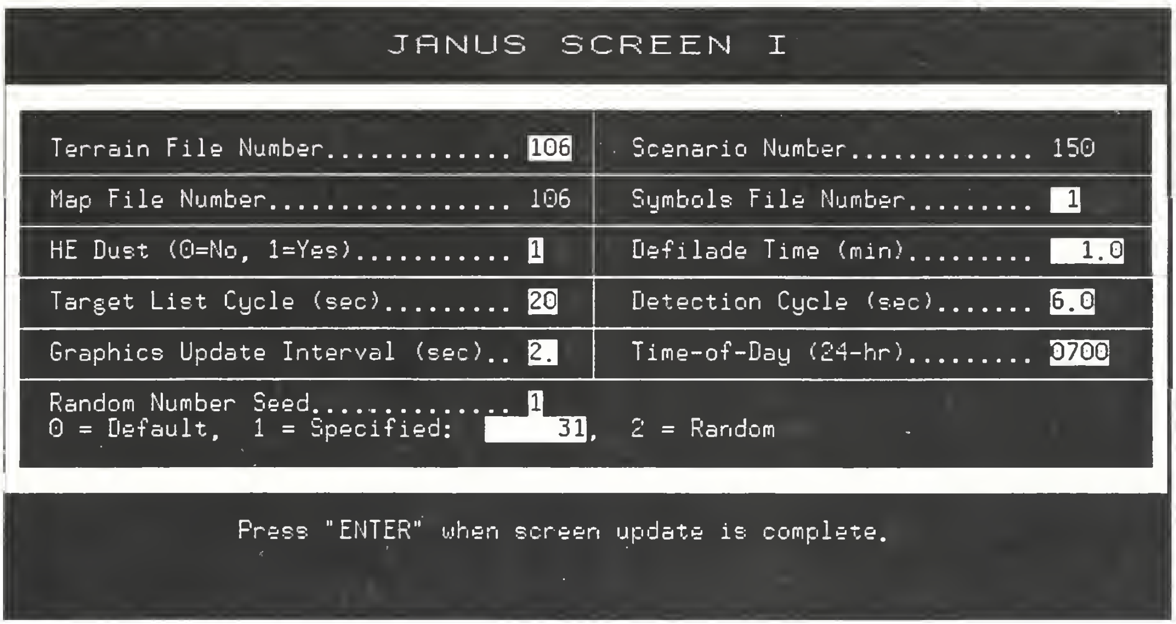

Figure A.1. JANUS Screen I showing the values of the target list update cycle time and the detection cycle times used in the scenarios run for this research.

APPENDIX B. JANUS SCREEN IV

The analyst may change the firing criterion on JANUS screen IV. Identification firing criterion is the most restrictive criterion while aimpoint is the least restrictive.

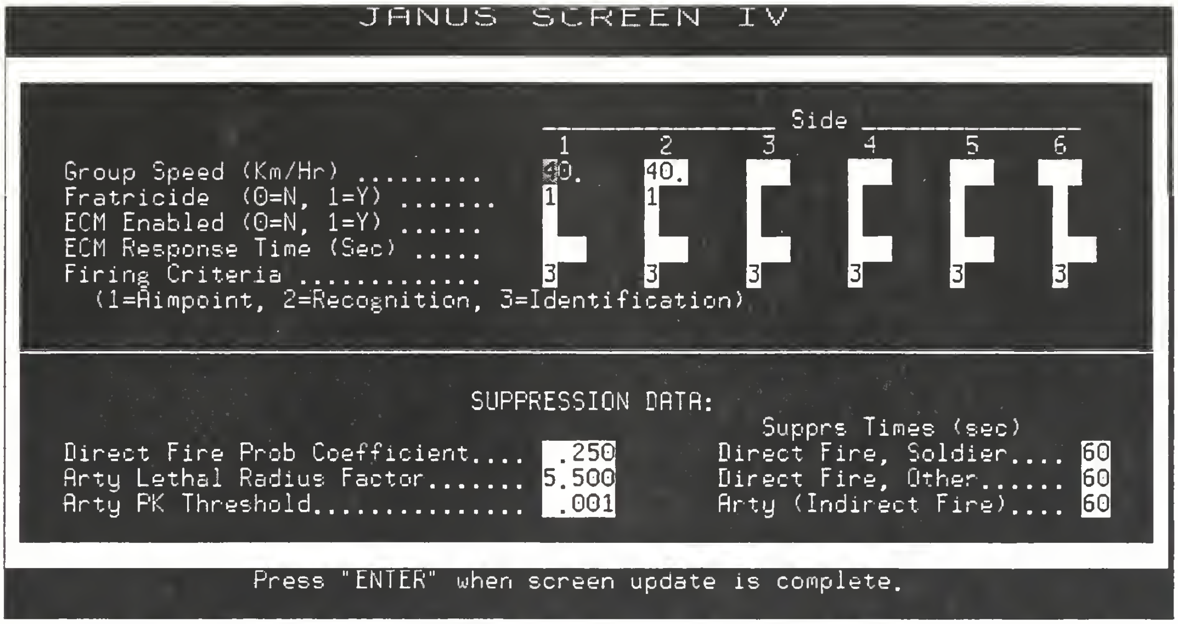

Figure B.1. JANUS screen IV showing how the user can select the firing criteria prior to running the JANUS scenarios

APPENDIX C. THE JAVA PROGRAM USED TO CALCULATE THE TARGET AVAILABILITY VALUES FROM THE MODIFIED DETECTION FILES

This program calculates the rates of the three-state Markov chain model of target acquisition and computes the steady-state values for target availability of all weapon systems. The target availability information is displayed in a frame on the screen. This information is then manually imported into an EXCEL spreadsheet for each weapon system combination for future use in MLE calculations.

```
import java.util.Vector;
import java.awt.*;
import javax.swing.*;
import javax.swing.table.*;
import javax.swing.SwingConstants;
import java.awt.event.*;
import java.util.Enumeration;
import java.util.StringTokenizer;
import java.io.FileOutputStream;
import java.io.FileInputStream;
import mil.army.trac.konig.*;

/*
 *Identification Firing Criterion
 *
 */

public class DataGraph11 {

  static String nodesFileName = new String("SYSMAP15037.DAT");
  static String arcsFileName = new String("DETECT15049.DAT");
  static String arcsOutFileName = new String("ArcsOut.data");
  static String nodesOutFileName = new String("NodesOut.data");

  public static void main(String[] args) {

    Graph originalDataGraph = new Graph();
    originalDataGraph.setProperty(Graph.nameKey,"Ziya Data Graph");

    try {
      originalDataGraph.inputNodes(new FileInputStream(nodesFileName));
      originalDataGraph.inputArcs(new FileInputStream(arcsFileName));
    }
    catch ( Exception e ) {
      System.err.println(e);
```

```
      e.printStackTrace();
      System.exit(1);
     }

    JTabbedPane myPane = new JTabbedPane();
    myPane.addTab("Original Data", new GraphPanel(originalDataGraph));

    System.out.println("Data Read");

    Graph tempGraph = new Graph(originalDataGraph);
    Graph transitionGraph = new Graph();
    transitionGraph.setProperty(Graph.nameKey,"Transition");
    while (tempGraph.getSize() > 0) {
      Enumeration e=tempGraph.arcs();
      Arc next = (Arc)e.nextElement();
      Node fromNode = next.getFromNode();
      Node toNode = next.getToNode();
      ArcSet common = new ArcSet();
      common.add(next);
        while (e.hasMoreElements()) {
          next = (Arc)e.nextElement();
          if ( (fromNode.equals(next.getFromNode())) &&
            (toNode.equals(next.getToNode())) ) {
            common.add(next);
          }
        }
      for (Enumeration e2=common.elements(); e2.hasMoreElements(); ) {
        next = (Arc)e2.nextElement();
        tempGraph.removeArc(next);
      }
      common.sort(new DoublePropertyComparator("TIME_ACQ"));

      double lastTime = 0.0;
      int lastLevel = 0;
      for (Enumeration e3=common.elements(); e3.hasMoreElements(); ) {
        next = (Arc)e3.nextElement();
        double nextTime = ((Number)next.getProperty("TIME_ACQ")).doubleValue();
        Arc newArc = new Arc(next.getFromNode(),next.getToNode());
        newArc.setProperty("TRANS_TIME", new Double(nextTime-lastTime));
        lastTime = nextTime;
        int nextLevel = ((Number)next.getProperty("DISCR_LEVEL")).intValue();
        String type = null;
        if (lastLevel==0) {
          switch (nextLevel) {
            case 1: case 2: type = new String("Nu");
            break;
            case 3: type = new String("Tau");
            break;
          }
        }
        if (lastLevel==1) {
          switch (nextLevel) {
```

```
          case 0: case 4: type = new String("Mu");
          break;
          case 3: type = new String("Lambda");
          break;
        }
      }
      if (lastLevel==2) {
        switch (nextLevel) {
          case 0: case 4: type = new String("Mu");
          break;
                          case 3: type = new String ("Lambda");
        }
      }
      if (lastLevel==3) {
        switch (nextLevel) {
          case 0: case 4: type = new String("MuOne");
          break;
        }
      }
      if (lastLevel==4) {
        switch (nextLevel) {
          case 1: case 2: type = new String("Nu");
          break;
          case 3: type = new String("Tau");
          break;
        }
      }
      if (type!=null) {
       newArc.setProperty("TRANS_TYPE",type);
       transitionGraph.addArc(newArc);
      }
      lastLevel = nextLevel;
    }//for e.3
  }//first while
myPane.addTab("Transition", new GraphPanel(transitionGraph));

System.out.println("TRANSITION Graph Built");

Graph sysGraph = new Graph();
sysGraph.setProperty(Graph.nameKey,"System");

for (Enumeration e=transitionGraph.arcs(); e.hasMoreElements(); ) {
  Arc nextArc = (Arc)e.nextElement();
  Node fromNode = nextArc.getFromNode();
  String sysName = (String)fromNode.getProperty("UNIT_NAME");
  Node fromSysNode = sysGraph.getNode(Node.nameKey,sysName);
  if (fromSysNode==null) {
    fromSysNode = new Node();
    fromSysNode.setProperty(Node.nameKey,sysName);
    sysGraph.addNode(fromSysNode);
  }
```

```
    Node toNode = nextArc.getToNode();
    sysName = (String)toNode.getProperty("UNIT_NAME");
    Node toSysNode = sysGraph.getNode(Node.nameKey,sysName);
    if (toSysNode==null) {
      toSysNode = new Node();
      toSysNode.setProperty(Node.nameKey,sysName);
      sysGraph.addNode(toSysNode);
    }

    String transType = (String)nextArc.getProperty("TRANS_TYPE");
    Number transTime = (Number)nextArc.getProperty("TRANS_TIME");
    Arc transArc = null;
    for (Enumeration eSys = sysGraph.arcs(); eSys.hasMoreElements(); ) {
       Arc next = (Arc)eSys.nextElement();
       if (fromSysNode.equals(next.getFromNode()) &&
          toSysNode.equals(next.getToNode()) &&
          next.getProperty("TRANS_TYPE").equals(transType) ) {
         transArc = next;
         break;
       }
    }
    if (transArc==null) {
      transArc = new Arc(fromSysNode,toSysNode);
      transArc.setProperty("TRANS_TYPE",transType);
      transArc.setProperty("transCount",new Integer(1));
      transArc.setProperty("transTotal",transTime);
            transArc.setProperty("transTypeValue",new Double(1/transTime.doubleValue()));
      sysGraph.addArc(transArc);
    }
    else {
      int count = ((Number)transArc.getProperty("transCount")).intValue()
               + 1;
      double total = ((Number)transArc.getProperty("transTotal")).doubleValue()
                 + transTime.doubleValue();
      transArc.setProperty("transCount",new Integer(count));
      transArc.setProperty("transTotal",new Double(total));
                 transArc.setProperty("transTypeValue",new Double(count/total));
    }//else
  }//outer for loop

       myPane.addTab("System", new GraphPanel(sysGraph));

Enumeration n1 = sysGraph.nodes();
Graph availGraph = new Graph();
while (n1.hasMoreElements()) {
  Node fromNode = (Node)n1.nextElement();
  Enumeration n2 = sysGraph.nodes();
  while (n2.hasMoreElements()) {
    Node toNode = (Node)n2.nextElement();
    ArcSet availArcs = new ArcSet();
    for (Enumeration a1 = sysGraph.arcs(); a1.hasMoreElements(); ) {
      Arc nextArc = (Arc)a1.nextElement();
```

```
        if (fromNode==nextArc.getFromNode() && toNode==nextArc.getToNode()) {
          availArcs.add(nextArc);
        }
      } // Arcs
          double mu=0.0; double muOne=0.0; double nu=0.0;
          double tau=0.0; double lambda=0.0;
      for (Enumeration a1 = availArcs.elements(); a1.hasMoreElements(); ) {
        Arc nextArc = (Arc)a1.nextElement();
        String type = (String)nextArc.getProperty("TRANS_TYPE");
        double typeValue = ((Number)nextArc.getProperty("transTypeValue")).doubleValue();
            if (type.equals("Mu")){
            mu=typeValue;
                }
                if (type.equals("muOne")){
            muOne=typeValue;
                }
                if (type.equals("Lambda")){
            lambda=typeValue;
                }
                if (type.equals("Tau")){
            tau=typeValue;
                }
                if (type.equals("Nu")){
            nu=typeValue;
                }
      } // Arcs in Avail
      double tgtAvail = ((nu*lambda)+tau*(lambda+mu))/
                    ((tau+nu+mu)*(lambda+mu));

      Arc availArc = new Arc(fromNode,toNode);
      availArc.setProperty("tgtAvail",new Double(tgtAvail));
      availGraph.addArc(availArc);
      System.out.println("end of adding arc "+availArc);
    } // toNodes
  } // fromNodes

  myPane.addTab("Availability",new GraphPanel(availGraph));

  GraphFrame theFrame = new GraphFrame("Ziya's Janus Data For Identification Firing Criterion");
  theFrame.getContentPane().add(myPane);
  theFrame.setVisible(true);

  } //main
} // DataGraph
```

For recognition firing criterion, only the following changes, concerning the transition rates in the Markov chain model, should be made to the program above:

```
String type = null;
```

```
    if (lastLevel==0) {
      switch (nextLevel) {
        case 1: type = new String("Nu");
        break;
        case 2: case 3: type = new String("Tau");
        break;
      }
    }
    if (lastLevel==1) {
      switch (nextLevel) {
        case 0: case 4: type = new String("Mu");
        break;
        case 2: case 3: type = new String("Lambda");
        break;
      }
    }
    if (lastLevel==2) {
      switch (nextLevel) {
        case 0: case 4: type = new String("MuOne");
        break;
      }
    }
    if (lastLevel==3) {
      switch (nextLevel) {
        case 0: case 4: type = new String("MuOne");
        break;
      }
    }
    if (lastLevel==4) {
      switch (nextLevel) {
        case 1: type = new String("Nu");
        break;
        case 2: case 3: type = new String("Tau");
        break;
      }
    }
    if (type!=null) {
     newArc.setProperty("TRANS_TYPE",type);
     transitionGraph.addArc(newArc);
    }
    lastLevel = nextLevel;
  }//for e.3
}//first while
myPane.addTab("Transition", new GraphPanel(transitionGraph));
System.out.println("TRANSITION Graph Built");
```

APPENDIX D. TARGET AVAILABILITY VALUES OBTAINED FOR EACH JANUS SCENARIO RUN

The following are the target availability values are obtained from the program in Appendix C. This program calculates the steady state values, $p_{VA}(\infty)$, of the continuous-time, three-state Markov chain model of target acquisition. These values correspond to the target availability parameters as explained in Chapter IV. The NaN values represent the cases where the steady state equation provides an undefined (zero divided by zero) value. This is normal since some of the weapon systems do not get detected enough times to have positive transition rates in the Markov chain model. This is due to the manuever plan selected, where those forces in the rear of the attack formation do not get detected enough times or do not make enough detections of the opposing side. The NaN values are not included in the averaged final target availability values.

Firing Criterion : Recognition Run: 15028

Detected Units		**Observing Units**		
	T-72	***T-80***	***BMP***	***BMP-2***
M1A1	0.1526	0.1353	0.0784	0.0399
M2IFV	1	NaN	0.1375	NaN

Firing Criterion : Recognition Run : 15028

Detected Units		**Observing Units**
	M1A1	***M2IFV***
T-72	0.088	0.3102
T-80	0.2035	0.9054
BMP	0.0962	0.1755
BMP-2	0.295	NaN

Firing Criterion : Recognition Run: 15029

Detected Units		**Observing Units**		
	T-72	***T-80***	***BMP***	***BMP-2***
M1A1	0.0552	0.2998	0.123	0.0589
M2IFV	0.4205	0.0152	0.1824	0.0526

Firing Criterion : Recognition Run : 15029

Detected Units		**Observing Units**
	M1A1	***M2IFV***
T-72	0.2181	0.7045
T-80	0.0667	NaN
BMP	0.0905	0.2148
BMP-2	0.2394	0.206

Firing Criterion : Recognition Run: 15030

Detected Units	T-72	T-80	BMP	BMP-2
M1A1	0.3198	0.0658	0.2143	NaN
M2IFV	1	NaN	NaN	NaN

Firing Criterion : Recognition Run : 15030

Detected Units	M1A1	M2IFV
T-72	0.6066	0.0559
T-80	0.3416	0.7122
BMP	0.1838	0.1601
BMP-2	0.0415	0.2605

Firing Criterion : Recognition Run: 15031

Detected Units	T-72	T-80	BMP	BMP-2
M1A1	0.1108	1	0.2055	0.0337
M2IFV	0.9999	0.2352	0.13	0.0858

Firing Criterion : Recognition Run : 15031

Detected Units	M1A1	M2IFV
T-72	0.3188	0.2463
T-80	NaN	NaN
BMP	0.0338	0.0866
BMP-2	0.5227	0.0561

Firing Criterion : Recognition Run: 15032

Detected Units	T-72	T-80	BMP	BMP-2
M1A1	0.0549	0.0718	0.0655	0.0881
M2IFV	0.3786	0.2696	0.0548	0.0233

Firing Criterion : Recognition Run : 15032

Detected Units	M1A1	M2IFV
T-72	0.825	NaN
T-80	0.265	0.0471
BMP	0.0973	0.1152
BMP-2	0.2458	0.0673

Firing Criterion : Recognition Run: 15033

Detected Units	T-72	T-80	BMP	BMP-2
M1A1	0.2481	0.1465	0.1021	0.1635
M2IFV	0.163	0.9999	0.1572	0.0503

Firing Criterion : Recognition Run : 15033

Detected Units	M1A1	M2IFV
T-72	0.5491	0.0427
T-80	0.0289	0
BMP	0.0332	0.2528
BMP-2	0.4419	0.0783

Firing Criterion : Recognition Run: 15034

Detected Units	T-72	T-80	BMP	BMP-2
M1A1	0.3674	0.0905	0.1837	NaN
M2IFV	0.4989	0.3058	0.0197	0.031

Firing Criterion : Recognition Run : 15034

Detected Units	M1A1	M2IFV
T-72	0.515	0.7541
T-80	0.083	0
BMP	0.1244	0.1726
BMP-2	0.2273	0.2458

Firing Criterion : Recognition Run: 15035

Detected Units	Observing Units			
	T-72	*T-80*	*BMP*	*BMP-2*
M1A1	0.7362	0.1641	0.2443	0.0121
M2IFV	0.6455	0.027	NaN	NaN

Firing Criterion : Recognition Run : 15035

Detected Units	Observing Units	
	M1A1	*M2IFV*
T-72	0.2468	0.2019
T-80	1	0.3906
BMP	0.1706	0.1853
BMP-2	0.0674	0.3305

Firing Criterion : Recognition Run: 15036

Detected Units	Observing Units			
	T-72	*T-80*	*BMP*	*BMP-2*
M1A1	0.3264	0.1201	0.4489	0.057
M2IFV	0.5073	0.1236	0.017	0.0394

Firing Criterion : Recognition Run : 15036

Detected Units	Observing Units	
	M1A1	*M2IFV*
T-72	0.6811	0.582
T-80	0.0855	0.3014
BMP	1	0.1556
BMP-2	0.3107	0.0603

Firing Criterion : Recognition Run: 15037

Detected Units	Observing Units			
	T-72	*T-80*	*BMP*	*BMP-2*
M1A1	0.2234	0.279	0.1351	NaN
M2IFV	0.2212	0.0147	NaN	0.9999

Firing Criterion : Recognition Run : 15037

Detected Units	Observing Units	
	M1A1	*M2IFV*
T-72	0.8126	0.4963
T-80	0.048	0.0164
BMP	0.0702	0.7102
BMP-2	0.3182	0.3301

FINAL Aij VALUES
Firing Criterion : Recognition

Detected Units	Observing Units			
	T-72	*T-80*	*BMP*	*BMP-2*
M1A1	0.25948	0.23729	0.18008	0.0647429
M2IFV	0.58349	0.248875	0.0998	0.1831857

FINAL Bji VALUES
Firing Criterion : Recognition

Detected Units	Observing Units	
	M1A1	*M2IFV*
T-72	0.48611	0.3771
T-80	0.2358	0.2966375
BMP	0.19	0.22287
BMP-2	0.27099	0.1816556

Firing Criterion : Identification Run: 15040

Detected Units	Observing Units T-72	T-80	BMP	BMP-2
M1A1	0.1157	0.1764	0.0988	0.0403
M2IFV	0.5317	0.1586	0.354	0.4911

Firing Criterion : Identification Run : 15040

Detected Units	Observing Units M1A1	M2IFV
T-72	0.5166	0.1755
T-80	0.5087	0.4458
BMP	0.1961	0.5558
BMP-2	0.5782	0.4967

Firing Criterion : Identification Run: 15041

Detected Units	Observing Units T-72	T-80	BMP	BMP-2
M1A1	0.3832	0.1137	0.1021	0.1664
M2IFV	0.1145	0.472	0.2832	0.0579

Firing Criterion : Identification Run : 15041

Detected Units	Observing Units M1A1	M2IFV
T-72	0.6271	0.5676
T-80	0.4148	0.3754
BMP	0.0831	0.2564
BMP-2	0.3129	0

Firing Criterion : Identification Run: 15042

Detected Units	Observing Units T-72	T-80	BMP	BMP-2
M1A1	0.4718	0.0549	0.1454	0.141
M2IFV	0.9999	0.0876	0.3349	0.0166

Firing Criterion : Identification Run : 15042

Detected Units	Observing Units M1A1	M2IFV
T-72	0.6275	0.1813
T-80	1	0.7461
BMP	0.1762	0.0586
BMP-2	0.705	0.2639

Firing Criterion : Identification Run: 15043

Detected Units	Observing Units T-72	T-80	BMP	BMP-2
M1A1	0.5089	0.502	0.5759	0.0818
M2IFV	0.5246	0.0222	0.334	1

Firing Criterion : Identification Run : 15043

Detected Units	Observing Units M1A1	M2IFV
T-72	0.5176	0.3242
T-80	0.1477	0.3119
BMP	0.0978	0.3619
BMP-2	0.5573	0.3318

Firing Criterion : Identification Run: 15044

Detected Units	Observing Units T-72	T-80	BMP	BMP-2
M1A1	0.3419	0.0278	0.2323	0.2202
M2IFV	0.2339	0.4091	0.084	0.0747

Firing Criterion : Identification Run : 15044

Detected Units	Observing Units M1A1	M2IFV
T-72	0.6086	0.4165
T-80	0.4278	0
BMP	0.096	0
BMP-2	0.4823	0.2295

Firing Criterion : Identification Run: 15045

Detected Units	*T-72*	*T-80*	*BMP*	*BMP-2*
M1A1	0.4307	0.5308	0.2646	0.3538
M2IFV	0.3031	0.6428	0.2523	0.3771

Observing Units

Firing Criterion : Identification Run : 15045

Detected Units	*M1A1*	*M2IFV*
T-72	0.6453	0.1954
T-80	0.0476	0
BMP	0.647	0
BMP-2	0.4746	0.8772

Observing Units

Firing Criterion : Identification Run: 15046

Detected Units	*T-72*	*T-80*	*BMP*	*BMP-2*
M1A1	0.466	0.1561	0.4362	0.199
M2IFV	0.9999	0.7346	0.3021	0.044

Observing Units

Firing Criterion : Identification Run : 15046

Detected Units	*M1A1*	*M2IFV*
T-72	0.8801	0.5184
T-80	0.3975	0.0742
BMP	0.0539	0.4549
BMP-2	0.2453	0.0514

Observing Units

Firing Criterion : Identification Run: 15047

Detected Units	*T-72*	*T-80*	*BMP*	*BMP-2*
M1A1	0.4386	0.0954	0.3552	0.0284
M2IFV	0.3328	0.5198	0.1664	0.632

Observing Units

Firing Criterion : Identification Run : 15047

Detected Units	*M1A1*	*M2IFV*
T-72	0.6061	0.3814
T-80	0.2438	0
BMP	0.1451	0.6391
BMP-2	0.719	0.3565

Observing Units

Firing Criterion : Identification Run: 15048

Detected Units	*T-72*	*T-80*	*BMP*	*BMP-2*
M1A1	0.2406	0.1217	0.4448	0.1289
M2IFV	0.6993	0.0497	0.2125	0.3079

Observing Units

Firing Criterion : Identification Run : 15048

Detected Units	*M1A1*	*M2IFV*
T-72	0.643	0
T-80	0.1366	0.0957
BMP	0.1686	0.1451
BMP-2	0.6175	0.4047

Observing Units

Firing Criterion : Identification Run: 15049

Detected Units	*T-72*	*T-80*	*BMP*	*BMP-2*
M1A1	0.0604	NaN	0.3669	0.0602
M2IFV	0.2956	1	0.0398	NaN

Observing Units

Firing Criterion : Identification Run : 15049

Detected Units	*M1A1*	*M2IFV*
T-72	0.6225	0.3032
T-80	0.0225	0.0424
BMP	NaN	0.8949
BMP-2	0.1851	0

Observing Units

FINAL Aij VALUES
Firing Criterion : Identification

Detected		**Observing Units**		
Units	***T-72***	***T-80***	***BMP***	***BMP-2***
M1A1	0.34578	0.1976444	0.30222	0.142
M2IFV	0.50353	0.40964	0.23632	0.3334778

FINAL Bji VALUES
Firing Criterion : Identification

Detected		**Observing Units**
Units	***M1A1***	***M2IFV***
T-72	0.62944	0.30635
T-80	0.3347	0.20915
BMP	0.1848667	0.33667
BMP-2	0.48772	0.30117

APPENDIX E. THE JAVA PROGRAM USED IN EXTRACTING NECESSARY INFORMATION FROM DIRECT-FIRE KILL FILES

The following JAVA program reads in the information contained in the modified JANUS direct-fire kill files. It extracts the information that is required for use in calculation for the conditional kill rate MLEs. The data that is extracted includes the times between kill events and the numbers of remaining entities at those times. This information is formatted and written to a text file. The text file is then imported into an EXCEL spreadsheet to obtain the attrition-rate estimates for each run of Scenario 150 in Appendix F.

```
import java.util.Vector;
import java.awt.*;
import javax.swing.*;
import javax.swing.table.*;
import javax.swing.SwingConstants;
import java.awt.event.*;
import java.util.Enumeration;
import java.util.StringTokenizer;
import java.io.FileOutputStream;
import java.io.FileInputStream;
import mil.army.trac.konig.*;
import java.text.DecimalFormat;
import java.io.*;

/**
 * Program Calculating Kill Rates
 * For Recognition and Identification
 * Firing Criteria
 */

public class DataGraph4 {

  static String nodesFileName = new String("SYSMAP15037.DAT");
  static String arcsFileName = new String("DFKILL15049.DAT");
  static String arcsOutFileName = new String("ArcsOut.data");
  static String nodesOutFileName = new String("NodesOut.data");

  public static void main(String[ ] args) {

System.out.println("start");
        int countM1A1 = 56;
```

```
        int countM2IFV = 56;
    int countT72 = 8;
        int countT80 = 8;
        int countBMP = 8;
        int countBMP2 = 8;

    int numBlueSys = 2;
    int numRedSys = 4;

    int [ ][ ] blueTotalKills = new int [numBlueSys][numRedSys];
    int [ ][ ] redTotalKills = new int [numRedSys][numBlueSys];

    for (int r=0; r<2; r++){
      for (int t=0; t<4; t++){
            blueTotalKills [r][t] = 0;
            redTotalKills [t][r] = 0;
      }
    }

    double alphaij [ ][ ] = new double [numBlueSys][numRedSys];
    double betaji [ ][ ] = new double [numRedSys][numBlueSys];

    for (int a=0; a<2; a++){
            for (int b=0; b<4; b++){
              alphaij [a][b] = 0;
              betaji [b][a] = 0;
            }
    }

System.out.println("data initialized");

    Graph killDataGraph = new Graph();
    killDataGraph.setProperty(Graph.nameKey,"Ziya DF KILL Data");

    try {
      killDataGraph.inputNodes(new FileInputStream(nodesFileName));
      killDataGraph.inputArcs(new FileInputStream(arcsFileName));
    }
    catch ( Exception e ) {
      System.err.println(e);
      e.printStackTrace();
      System.exit(1);
    }


System.out.println("Data Read");

    Graph tempGraph = new Graph(killDataGraph);
    //Graph killGraph = new Graph();
    //killGraph.setProperty(Graph.nameKey,"killGraph");

    Vector survivorsM1A1 = new Vector();
```

```
Vector survivorsM2IFV = new Vector();
Vector survivorsT72 = new Vector();
Vector survivorsT80 = new Vector();
Vector survivorsBMP = new Vector ();
Vector survivorsBMP2 = new Vector ();
      Vector v = new Vector();

      survivorsM1A1.addElement(new Integer(56));
      survivorsM2IFV.addElement(new Integer(56));
      survivorsT72.addElement(new Integer(8));
      survivorsT80.addElement(new Integer(8));
      survivorsBMP.addElement(new Integer(8));
      survivorsBMP2.addElement(new Integer(8));

      Enumeration m = tempGraph.arcs();
while (m.hasMoreElements()){
  Arc next = (Arc) m.nextElement();
          Node fromNode = next.getFromNode();
          Node toNode = next.getToNode();
          String firerName = ((String) fromNode.getProperty("UNIT_NAME"));
          String victimName = ((String) toNode.getProperty("UNIT_NAME"));

      //ALPHAij stuff
//T-72 vs. M1A1
  if ((firerName.equals("T-72"))&& (victimName.equals("M1A1"))){
          countM1A1= countM1A1-1;
    survivorsM1A1.addElement(new Integer(countM1A1));
    blueTotalKills[0][0] = blueTotalKills[0][0] +1;
        }

        else if ((firerName.equals("T-80"))&& (victimName.equals("M1A1"))){
    countM1A1= countM1A1 -1;
    survivorsM1A1.addElement(new Integer(countM1A1));
          blueTotalKills[0][1] = blueTotalKills[0][1] +1;
  }

        else if ((firerName.equals("BMP"))&& (victimName.equals("M1A1"))){
          countM1A1 = countM1A1 -1;
    survivorsM1A1.addElement(new Integer(countM1A1));
          blueTotalKills[0][2] = blueTotalKills[0][2] +1;
  }

  else if ((firerName.equals("BMP-2"))&& (victimName.equals("M1A1"))){
          countM1A1 = countM1A1 -1;
    survivorsM1A1.addElement(new Integer(countM1A1));
          blueTotalKills[0][3] = blueTotalKills[0][3] +1;
  }
  else if (!victimName.equals("M1A1")){
          survivorsM1A1.addElement(survivorsM1A1.lastElement());
        }

  //T-72 vs. M2IFV
```

```
if ((firerName.equals("T-72"))&& (victimName.equals("M2IFV"))){
        countM2IFV = countM2IFV-1;
  survivorsM2IFV.addElement(new Integer(countM2IFV));
        blueTotalKills[1][0] = blueTotalKills[1][0] +1;
      }

      else if ((firerName.equals("T-80"))&& (victimName.equals("M2IFV"))){
  countM2IFV = countM2IFV-1;
  survivorsM2IFV.addElement(new Integer(countM2IFV));
        blueTotalKills[1][1] = blueTotalKills[1][1] +1;
      }

      else if ((firerName.equals("BMP"))&& (victimName.equals("M2IFV"))){
        countM2IFV= countM2IFV -1;
  survivorsM2IFV.addElement(new Integer(countM2IFV));
        blueTotalKills[1][2] = blueTotalKills[1][2] +1;
      }

      else if ((firerName.equals("BMP-2"))&& (victimName.equals("M2IFV"))){
        countM2IFV = countM2IFV -1;
  survivorsM2IFV.addElement(new Integer(countM2IFV));
        blueTotalKills[1][3] = blueTotalKills[1][3] +1;
      }
else if (!victimName.equals("M2IFV")){
survivorsM2IFV.addElement(survivorsM2IFV.lastElement());
        }

      //BETAji stuff
//M1A1 vs. T-72

if ((firerName.equals("M1A1"))&& (victimName.equals("T-72"))){
        countT72= countT72-1;
  survivorsT72.addElement(new Integer(countT72));
        redTotalKills[0][0] = redTotalKills[0][0] +1;
      }

      else if ((firerName.equals("M2IFV"))&& (victimName.equals("T-72"))){
        countT72 =countT72-1;
  survivorsT72.addElement(new Integer(countT72));
        redTotalKills[0][1] = redTotalKills[0][1] +1;

      }
else if (!victimName.equals("T-72")){
  survivorsT72.addElement(survivorsT72.lastElement());
}

//M1A1 vs. T-80

if ((firerName.equals("M1A1"))&& (victimName.equals("T-80"))){
  countT80 = countT80-1;
  survivorsT80.addElement(new Integer(countT80));
```

```
            redTotalKills[1][0] = redTotalKills[1][0] +1;
          }
    else if ((firerName.equals("M2IFV"))&& (victimName.equals("T-80"))){
            countT80 = countT80-1;
      survivorsT80.addElement(new Integer(countT80));
            redTotalKills[1][1] = redTotalKills[1][1] +1;
          }
    else if (!victimName.equals("T-80")){
      survivorsT80.addElement(survivorsT80.lastElement());
    }

    //M1A1 vs. BMP

    if ((firerName.equals("M1A1"))&& (victimName.equals("BMP"))){
      countBMP = countBMP-1;
      survivorsBMP.addElement(new Integer(countBMP));
            redTotalKills[2][0] = redTotalKills[2][0] +1;
          }
         else if ((firerName.equals("M2IFV"))&& (victimName.equals("BMP"))){
          countBMP = countBMP-1;
     survivorsBMP.addElement(new Integer(countBMP));
          redTotalKills[2][1] = redTotalKills[2][1] +1;
         }

    else if (!victimName.equals("BMP")){
      survivorsBMP.addElement(survivorsBMP.lastElement());
    }

         //M1A1 vs. BMP-2

         if ((firerName.equals("M1A1"))&& (victimName.equals("BMP-2"))){
           countBMP2 = countBMP2-1;
      survivorsBMP2.addElement(new Integer(countBMP2));
           redTotalKills[3][0] = redTotalKills[3][0] +1;
          }
    else if ((firerName.equals("M2IFV"))&& (victimName.equals("BMP-2"))){
          countBMP2=countBMP2-1;
     survivorsBMP2.addElement(new Integer(countBMP2));
          redTotalKills[3][1] = redTotalKills[3][1] +1;
          }
    else if (!victimName.equals("BMP-2")){
      survivorsBMP2.addElement(survivorsBMP2.lastElement());
    }

  }//while m.hasMoreElements

System.out.println("while loop complete");

  double lastTime = 0.0;
  int victimCount = 0;

  String lastVictimName = new String("");
```

```
for (Enumeration e3=tempGraph.arcs(); e3.hasMoreElements(); ) {
  Arc next = (Arc)e3.nextElement();
       Node fromNode1 = next.getFromNode();
       Node toNode1 = next.getToNode();
       String victimName = ((String)toNode1.getProperty("UNIT_NAME"));
       String firerName = ((String)fromNode1.getProperty("UNIT_NAME"));
  double nextTime = ((Number)next.getProperty("TIME_KILL")).doubleValue();
  Arc newArc = new Arc(next.getFromNode(),next.getToNode());
  newArc.setProperty("victimName",victimName);
  double timeBWKills = nextTime-lastTime;
  newArc.setProperty("TIME_BW_KILLS", new Double(timeBWKills));
       v.addElement(new Double(timeBWKills));
  victimCount++;
  newArc.setProperty("victim_Count",new Integer(victimCount));
  lastTime = nextTime;
  killDataGraph.addArc(newArc);
      }

JTabbedPane myPane = new JTabbedPane();
myPane.addTab("DF Kill Data", new GraphPanel(killDataGraph));
GraphFrame theFrame =new GraphFrame ("Ziya's JANUS DF Kill Data");
theFrame.getContentPane().add(myPane);
theFrame.setVisible(true);

Integer [ ] survM1A1 = new Integer [survivorsM1A1.size()];
Integer [ ] survM2IFV = new Integer [survivorsM2IFV.size()];
Integer [ ] survT72 = new Integer [survivorsT72.size()];
Integer [ ] survT80 = new Integer [survivorsT80.size()];
Integer [ ] survBMP = new Integer [survivorsBMP.size()];
Integer [ ] survBMP2 = new Integer [survivorsBMP2.size()];
      Double [ ] timeBWKills = new Double [v.size()];

survivorsM1A1.copyInto(survM1A1);
survivorsM2IFV.copyInto(survM2IFV);
survivorsT72.copyInto(survT72);
survivorsT80.copyInto(survT80);
survivorsBMP.copyInto(survBMP);
survivorsBMP2.copyInto(survBMP2);
      v.copyInto(timeBWKills);

System.out.println(survM1A1.length);
System.out.println(survM2IFV.length);
      System.out.println(survT72.length);
      System.out.println(survT80.length);
      System.out.println(survBMP.length);
      System.out.println(survBMP2.length);
      System.out.println("v"+timeBWKills.length);

      DecimalFormat df = new DecimalFormat("#.0000");

      try {
        FileOutputStream outputFile = new FileOutputStream("outPut");
```

```
            PrintWriter printOut= new PrintWriter(outputFile);

            printOut.write("\n"+"DF Kill Scenario 15040.DAT \n\n");
            printOut.checkError();
        printOut.write("M1A1"+"\t"+"M2IFV" +           "\t"
+"T72"+"\t"+"T80"+"\t"+"BMP"+"\t"+"BMP2"+"\t"+"TimeBWKills"+"\n\n");
            printOut.checkError();

          for (int i=0;i<survBMP2.length;i++){
            String s = (survM1A1[i].intValue())+"\t"
            +(survM2IFV[i].intValue())+"\t"
            +(survT72[i].intValue())+"\t"
            +(survT80[i].intValue())+"\t"
            +(survBMP[i].intValue())+"\t"
            +(survBMP2[i].intValue() +"\t"+ df.format((timeBWKills[i].doubleValue()))+"\n");

            printOut.print(s);
            printOut.checkError();
    }
          outputFile.close();
          }
          catch (IOException e) {
            System.err.println(e);
          }

  }//main
}//DataGraph
```

APPENDIX F. THE ATTRITION-RATE ESTIMATES CALCULATED USING THE MAXIMUM LIKELIHOOD ESTIMATION

Attrition-Rate Coefficients' (Conditional Kill Rates) Values Obtained
From Scenarios Run With Identification Firing Criterion

Run: 15040

ALPHAij	T-72	T-80	BMP	BMP-2	BETAji	M1A1	M2IFV
M1A1	0.0806	0.0536	0.0337	0.0455	T-72	0.2099	0.0000
M2IFV	0.0645	0.0459	0.0472	0.0228	T-80	0.0011	0.0034
					BMP	0.0000	0.0000
					BMP-2	0.5416	0.0215

Run : 15041

ALPHAij	T-72	T-80	BMP	BMP-2	BETAji	M1A1	M2IFV
M1A1	0.0875	0.0366	0.0424	0.0257	T-72	0.1937	0.0000
M2IFV	0.0389	0.0146	0.0777	0.0385	T-80	0.0012	0.0080
					BMP	0.0000	0.0000
					BMP-2	0.3936	0.0000

Run 15042

ALPHAij	T-72	T-80	BMP	BMP-2	BETAji	M1A1	M2IFV
M1A1	0.0758	0.0447	0.0349	0.0413	T-72	0.1527	0.0014
M2IFV	0.0505	0.0224	0.0627	0.0310	T-80	0.0012	0.0056
					BMP	0.0000	0.0000
					BMP-2	0.2497	0.1168

Run 15043

ALPHAij	T-72	T-80	BMP	BMP-2	BETAji	M1A1	M2IFV
M1A1	0.1062	0.0275	0.0334	0.0160	T-72	0.0795	0.0000
M2IFV	0.0398	0.0275	0.0534	0.0561	T-80	0.0059	0.0000
					BMP	0.0000	0.0000
					BMP-2	0.2731	0.0458

Run 15044

ALPHAij	T-72	T-80	BMP	BMP-2	BETAji	M1A1	M2IFV
M1A1	0.1566	0.0450	0.0224	0.0420	T-72	0.1943	0.0013
M2IFV	0.0418	0.0375	0.0299	0.0252	T-80	0.0013	0.0000
					BMP	0.0000	0.0000
					BMP-2	0.6784	0.1199

Run 15045

ALPHAij	T-72	T-80	BMP	BMP-2	BETAji	M1A1	M2IFV
M1A1	0.1359	0.0327	0.0223	0.0536	T-72	0.1209	0.0000
M2IFV	0.0510	0.0245	0.0223	0.0429	T-80	0.0026	0.0000
					BMP	0.0000	0.0000
					BMP-2	0.1994	0.0877

Run 15046

ALPHAij	T-72	T-80	BMP	BMP-2	BETAji	M1A1	M2IFV
M1A1	0.0614	0.0899	0.0141	0.0502	T-72	0.1579	0.0014
M2IFV	0.0245	0.0300	0.0634	0.0502	T-80	0.0012	0.0000
					BMP	0.0000	0.0000
					BMP-2	0.1665	0.0000

Run 15047

ALPHAij	T-72	T-80	BMP	BMP-2	BETAji	M1A1	M2IFV
M1A1	0.1462	0.0233	0.0285	0.0120	T-72	0.1407	0.0014
M2IFV	0.0562	0.0156	0.0571	0.0599	T-80	0.0024	0.0000
					BMP	0.0000	0.0000
					BMP-2	0.3788	0.0445

Run 15048

ALPHAij	T-72	T-80	BMP	BMP-2	BETAji	M1A1	M2IFV
M1A1	0.1062	0.0778	0.0137	0.0082	T-72	0.1408	0.0000
M2IFV	0.0637	0.0233	0.0617	0.0164	T-80	0.0023	0.0000
					BMP	0.0000	0.0000
					BMP-2	0.9672	0.0615

Run 15049

ALPHAij	T-72	T-80	BMP	BMP-2	BETAji	M1A1	M2IFV
M1A1	0.1547	0.0461	0.0230	0.0293	T-72	0.1657	0.0000
M2IFV	0.0773	0.0230	0.0538	0.0097	T-80	0.0000	0.0000
					BMP	0.0000	0.0000
					BMP-2	0.2552	0.1614

Final Averaged Attrition-Rate
Coefficients For Identification

ALPHAij	T-72	T-80	BMP	BMP-2	BETAji	M1A1	M2IFV
M1A1	0.1111	0.0477	0.0268	0.0324	T-72	0.1556	0.0005
M2IFV	0.0508	0.0264	0.0529	0.0353	T-80	0.0019	0.0017
					BMP	0.0000	0.0000
					BMP-2	0.4104	0.0659

Attrition-Rate Coefficients' (Conditional Kill Rates) Values Obtained
From Scenarios Run With Recognition Firing Criterion

Run: 15028.DAT

ALPHAij	T-72	T-80	BMP	BMP-2	BETAji	M1A1	M2IFV
M1A1	0.0695	0.0725	0.0281	0.0402	T-72	0.0975	0.0000
M2IFV	0.1158	0.0322	0.0636	0.0390	T-80	0.0040	0.0000
					BMP	0.0000	0.0000
					BMP-2	0.3316	0.2081

Run 15029.DAT

ALPHAij	T-72	T-80	BMP	BMP-2	BETAji	M1A1	M2IFV
M1A1	0.0776	0.0572	0.0274	0.0153	T-72	0.0734	0.0000
M2IFV	0.0465	0.0163	0.0897	0.0738	T-80	0.0039	0.0000
					BMP	0.0000	0.0000
					BMP-2	0.2455	0.0922

Run 15030.DAT

ALPHAij	T-72	T-80	BMP	BMP-2	BETAji	M1A1	M2IFV
M1A1	0.1569	0.0618	0.0304	0.0576	T-72	0.0716	0.0015
M2IFV	0.0523	0.0177	0.0609	0.0556	T-80	0.0044	0.0000
					BMP	0.0000	0.0000
					BMP-2	0.4007	0.2227

Run 15031.DAT

ALPHAij	T-72	T-80	BMP	BMP-2	BETAji	M1A1	M2IFV
M1A1	0.1166	0.0478	0.0481	0.0566	T-72	0.0982	0.0000
M2IFV	0.0933	0.0159	0.0621	0.0409	T-80	0.0027	0.0000
					BMP	0.0000	0.0000
					BMP-2	0.2025	0.1307

Run 15032.DAT

ALPHAij	T-72	T-80	BMP	BMP-2	BETAji	M1A1	M2IFV
M1A1	0.1247	0.0300	0.0346	0.0567	T-72	0.1045	0.0000
M2IFV	0.0356	0.0300	0.0556	0.0682	T-80	0.0040	0.0000
					BMP	0.0000	0.0000
					BMP-2	0.3825	0.0000

Run 15033.DAT

ALPHAij	T-72	T-80	BMP	BMP-2	BETAji	M1A1	M2IFV
M1A1	0.1880	0.0391	0.0454	0.0494	T-72	0.0876	0.0000
M2IFV	0.0684	0.0078	0.0456	0.0592	T-80	0.0029	0.0000
					BMP	0.0000	0.0000
					BMP-2	0.4290	0.2802

Run 15034.DAT

ALPHAij	T-72	T-80	BMP	BMP-2	BETAji	M1A1	M2IFV
M1A1	0.1222	0.0595	0.0291	0.0717	T-72	0.0870	0.0018
M2IFV	0.0407	0.0085	0.0732	0.0551	T-80	0.0044	0.0000
					BMP	0.0000	0.0000
					BMP-2	0.1801	0.2120

Run 15035.DAT

ALPHAij	T-72	T-80	BMP	BMP-2	BETAji	M1A1	M2IFV
M1A1	0.1003	0.0297	0.0470	0.0000	T-72	0.1167	0.0000
M2IFV	0.0251	0.0223	0.0743	0.1296	T-80	0.0037	0.0025
					BMP	0.0000	0.0000
					BMP-2	0.2522	0.0751

Run 15036.DAT

ALPHAij	T-72	T-80	BMP	BMP-2	BETAji	M1A1	M2IFV
M1A1	0.1127	0.0519	0.0368	0.0662	T-72	0.0794	0.0000
M2IFV	0.0705	0.0259	0.0296	0.0639	T-80	0.0029	0.0052
					BMP	0.0000	0.0000
					BMP-2	0.3994	0.2493

Run 15037.DAT

ALPHAij	T-72	T-80	BMP	BMP-2	BETAji	M1A1	M2IFV
M1A1	0.1335	0.0513	0.0288	0.0483	T-72	0.0610	0.0000
M2IFV	0.0334	0.0171	0.0650	0.1089	T-80	0.0041	0.0000
					BMP	0.0000	0.0000
					BMP-2	0.4126	0.0000

Final Averaged Attrition-Rate
Coefficients For Recognition

ALPHAij	T-72	T-80	BMP	BMP-2	BETAji	M1A1	M2IFV
M1A1	0.1202	0.0501	0.0356	0.0462	T-72	0.0877	0.0003
M2IFV	0.0582	0.0194	0.0620	0.0694	T-80	0.0037	0.0008
					BMP	0.0000	0.0000
					BMP-2	0.3236	0.1470

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center..2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Professor James G. Taylor..1
   Code OR/Ta
   Naval Postgraduate School
   Monterey, California 93943-5102

4. CDR Ronald L. Brown...1
   Code OR/Br
   Naval Postgraduate School
   Monterey, California 93943-5102

5. MAJ William S. Murphy Jr..2
   Naval Postgraduate School
   P.O. Box 8692
   Monterey, California 93943-0692

6. Kara Kuvvetleri Komutanligi...1
   Personel Daire Bsk.ligi
   Bakanliklar, Ankara
   Turkey

7. Kara Harp Okulu Komutanligi...1
   Kutuphane
   06654 Bakanliklar, Ankara
   Turkey

8. Bilkent Universitesi Kutuphanesi...1
   06533 Bilkent, Ankara,
   Turkey

9. Orta Dogu Teknik Universitesi Kutuphanesi..1
Balgat, Ankara
Turkey

10. Bogazici Universitesi Kutuphanesi..1
Bebek, Istanbul
Turkey

11. 1LT. Ugur Ziya Yildirim..2
Kara Harp Okulu Ogretim Bsk.ligi
Yon. Bil. Bol.
06654 Bakanliklar, Ankara
Turkey